أصول البرمجة

بيسك BASIC

تأليف :

المهندس سامي كاظم حسن الرماحي

ماجستير هندسة سيطرة وأجهزة

الجامعة التكنولوجية ـ بغداد

دار الراتب الجامعية

أصول البرمجة

الاهـــداء

إلـــى والـــدي

محتويات الكتاب

الفصل الرابع : كيفية التعامل مع حاسبة الكترونية لتنفيذ برنامج بيسيك؟

تصدير

إذا أردت استخدام الحاسبة الالكترونية كأداة لحل الكثير من المسائل الاكاديمية أو التجارية أو البيتية ، فيجب أن تصمم خطوات دقيقة وتفصيلية لطريقة حل أي من تلك المسائل. تدعى تلك الطريقة بعد كتابتها بلغة برمجة مناسبة بالبرنامج (program) .

ومن لغات البرمجة الواسعة الانتشار والمتواجدة ضمن برامجيات جميع أنواع الحساسبات الالكترونية هي لغة بيسك (BASIC) والتي تتميز بسهولتها عند التعلم والاستخدام والتذكر. وهذا الكتاب هو لارشاد القارىء في أصول برمجة الحاسبة الالكترونية بلغة بيسك .

تم تصميم لغة بيسك في سنة (1964) من قبل العالمين كمني (Kemeny) وكيرتز (Kurtz) في جامعة دارموث (Dartmouth) . ومنذ ذلك الحين ولغة بيسك تثبت أنها لغة البرمجة الأكثر انتشاراً بحيث أصبح عدد الذين يتقنونها يفوق تعداد الذين يتقنون الكثير من بعض لغات البشر الحية والمعمول بها يومياً !

منذ إعلان لغة بيسك ، أكد العالمان أن بإمكان المبتديء تعلم لغة بيسك بثلاثة ساعات فقط! وهذا صحيح ، من خلال تجربتي مع هذه اللغة المحببة الى النفس.

كان هدفي عند كتابة «أصول برمجة الحاسبة الالكترونية بلغة بيسك» هو ترغيب القارىء لاكتساب المهارات في أمور ثلاثة ضرورية لاستخدام الحاسبة الالكترونية عند حل مسألة معينة. الأول: هو تعلّم القارىء كيفية برمجة حاسبة الكترونية، أي تعلّم أساليب كتابة طريقة (خوارزمية) حل أية مسألة متوجة بتعلم أسلوب البرمجة الهيكلية (Structured programming).

والأمر الثاني هو تعلم القارىء كيفية كتابة برنامج بيسك عام يمكن تنفيذه على أية حاسبة الكترونية متوفرة لدى القارىء، سواء كانت حاسبة الكترونية كبيرة الحجم أو متوسطة أو شخصية. وهذا ما يدعى بقابلية التحويل (portability) في علم الحاسبة الالكترونية. فعبارات لغة بيسك المشروحة في هذا الكتاب هي ما اتفقت عليه جميع نسخ هذه اللغة المتواجدة ضمن برامجيات معظم الحاسبات الالكترونية بأنواعها المختلفة.

والأمر الثالث هو تعلّم كيفية التعامل مع الحاسبة الالكترونية لتنفيذ برنامج مكتوب بلغة بيسك، وذلك بالتعرف على أوامر (Commands) لغة بيسك.

يتميز كتاب «أصول برمجة الحاسبة الالكترونية بلغة بيسك» بأن قراءته لا تتطلب أية معرفة مسبقة في علوم الحاسبة الالكترونية. فما على القارىء سوى دراسة الفقرة تلو الأخرى لتحقيق أهداف كل فصل، والتنقل من فصل إلى آخر لاستكمال متطلبات تعلمه واستخدامه للغة بيسك في حل الكثير من المسائل الأكاديمية والتجارية والبيتية. وقد وضعت ما بين الفقرات في كل فصل عدداً من الامتحانات القصيرة تحت عنوان «اختبر نفسك» ليستطيع القارىء معرفة مستوى فهمه لفقرات هذا الكتاب، ولجعل الكتاب بمجمله مفيداً للقارىء الذي يتعلم بمفرده (بدون معلم برمجة). وعند إكمال متطلبات أهداف كل فصل تطرح

عـدة مسـائـل محلولـة اقتبست من أوراق امتحــان هيئـات وجــامعـات عـالمية، ثم تجمّـع أهم أفكار الفصـل الجوهـرية كخـلاصـة، وتتبع بقـائمة لأهم الكلمـات التي قد تعلمهـا القارىء من جـراء دراسته لهـذا الفصل.

يختم كـل فصل بمجمـوعة تمـارين مسلية وضعت تحت عنـاوين أربعة هي :

«استخدم كلمات قد تعلمتها»
«صح أم خطأ»
«أجب باختصار»
«اختار الجواب الصحيح»

لنجعـل من عملية تـدقيق القارىء لمـدى إدراكه وفهمـه لمفـردات كـل فصل غير مملة وتثقيفية في الوقت نفسه.

يستطيع قراءة هذا الكتاب كل من لديه الرغبة في بـرمجة حـاسبته الالكتـرونية، وبغض النـظر عن تحصيله الدراسي. بحيث يستفيـد منـه كـل من: المهندس الممـارس الذي يـريد جعـل الحاسبـة الالكترونيـة تساعده في اتخاذ بعض القرارات، والطبيب المعالج الذي يريد الإفـادة من سرعة ودقة وكفاءة الحـاسبة الالكتـرونية عنـد تشخيص الأمراض أو وضع نظام معلومات في عيادته الخاصـة أو المستشفى يعينه كمـوسوعـة طبيـة مخزونـة عند اللزوم، بـالضغط فقط على بعض الأزرار، وكذلـك يستفيـد منـه المـوظف الـذي يعمــل في حقـل المحـاسبـة والتـدقيق والاقتصاد والمكتبات والتـدريس، والمدراء الـذين يريـدون الافادة من البرامجيات الكثيرة في حقل إدارة المعـامل والشـركات، كـذلك وضـع هذا الكتاب لطلاب الجامعـات الذين يـريدون التعـرف ولأول مرة على

لغة برمجة سهلة وفعالة تساعدهم في حل المسائل المعقـدة، ولطلبـة ما قبـل الدراسـة الجامعية كهواة يـريدون تنميـة قابليـاتهم الذهنية لصنـع مستقبل زاهر.

ومن الله التوفيق

سامي الرماحي

بغداد ـ ١٩٨٧/١٩٨٨

الفصل الأول

كيف تبرمج حاسبة الكترونية؟

1. المقدمة
2. البرمجـــة
   اختبر نفسك
3. خطوات حل مسألة باستخدام الحاسبة الالكترونية
4. الخوارزميـة
   اختبر نفسك
5. المخطط الانسيابي
   اختبر نفسك
6. شبـه الشفرة
7. البرمجـة الهيكلية
   اختبر نفسك
8. مسائل محـلولـة
9. خـلاصـــة
10. كلمات قد تعلمتها
11. تمارين (١)

1. مقدمة :

قد أشيع الكثير عن المقدرات الخيالية للحاسبة الالكترونية وإمكانياتها في حل الألغاز التي يعجز الإنسان عن فك رموزها. ولكن الحقيقة تكمن في أن الحاسبة الالكترونية تؤدي فقط ما يتم برمجته عليه. أي يجب ذكر الخطوات الدقيقة والتفصيلية لاجراءات حل أي مسألة حتى تستطيع هي من اتباع تلك الخطوات وبصرامة كبيرة الواحدة تلو الأخرى وبدون ملل أو كلل حتى تتوصل الى الجواب المبني على تنفيذ تلك الخطوات المصممة من قبل الإنسان. لذا تم تكريس هذا الفصل لتعليم القارىء «كيفية برمجة حاسبة الكترونية». ليستطيع استغلال أداة العصر العصر (الحاسبة الالكترونية) لانجاز أعماله بكفاءة ودقة وسرعة ممتازة.

سيهيء لك هذا الفصل بعض المفاهيم والمبادىء الأساسية التي لا بد منها لكل مبرمج لكي يخلق جواً من ضبط الجهد العقلي لتصميم برنامجاً يؤدي الهدف الذي وجد من أجله. بعد دراستك لهذا الفصل، عزيزي القارىء، سيكون بإمكانك:

أولاً : إدراك «ما هي البرمجة؟».

ثانياً : معرفة الخطوات الستة الواجب اتباعها عند استخدام الحاسبة

الالكترونية كأداة في حل المسائل.

ثالثاً : تقسيم وتنظيم حل المسألة التي تريد برمجتها.

رابعاً : تعريف الخوارزمية، شبه الشفرة، المخطط الانسيابي، البرمجة الهيكلية، تركيبات الأنماط المنطقية، انشوطة، عداد.

خامساً : معرفة كيفية كتابة خوارزمية حل مسألة تمهيداً لبرمجتها في حاسبة الكترونية.

2. البرمجــة :

من المؤكد أنه قد تبادر الى ذهنـك، «ما هي البـرمجة؟»، وأنت تهم بحل مسألة باستخدام حسابـة الكترونيـة. إن البرمجـة هي تغذيـة الحاسبـة الالكتـرونية بخـطوات دقيقة وتفصيليـة، والتي تمثل خـطوات تـوصلك الى حل مسألة معينة. أي أنها إخبـار الحاسبة الالكترونية «مـاذا تفعل» للتـوصل الى حـل. ولكن هذا لا يتم باستخـدام مفردات اللغـة التي اعتـدت على التحدث بها يومياً، بل يجب استخدام لغة برمجة خاصـة تستطيـع الحاسبـة الالكترونية تفسيرها ومن ثم تنفيذ أوامرها وعباراتها.

إن من المهمـات الشاقـة هو أن تعـرف «كيف تحـل مـا ستحـل» لأن ذلـك يتطلب تحليـلاً دقيقـاً للخـطوات التي ستتخـذهـا للوصـول الى حـل صحيح لمسألة معينة. وهذا ما تتطلبه برمجة الحـاسبة الالكتـرونية، أي من الضـروري أن تعـرف كيف قـد تـوصلت الى قـرار معين. فـإذا لم تستـطع وصف خـطوات حـل مسـألـة، فـإنـك لا تستـطيـع الإيعـاز الى الحـاسبــة الالكترونية لحل تلك المسألة، وبالتالي سوف لن تستطيع بـرمجة الحـاسبة الالكترونية لانجاز تلك المهمة.

تكـون البرمجـة مفيدة في الحـالات التي تعـرف فيهـا بـالضبط كيف نتـوصل الى الإجـابـة ولكن من الصعب أو المضجـر الـوصـول الى الحـل

النهـائي. مثلاً، هنـالك مسـائل تتـطلب التكرار الممـل لعمليـات ريـاضيـة معقـدة تتم على أرقام كبيـرة، وبالتـالي التـدقيق المستمـر لخـطوات الحـل السـابقة لكي نتجنب الأخـطاء الصغيرة لنـأخذ بنـظر الاعتبار عمليـة ضـرب العـدد (627 153 8) في العدد (862 935 4)، حيث يستـطيع أي طـالب له معـرفة بجـدول الضرب أن ينجـزها ولكن سيتـطلب ذلك الـوقت والتـدقيق المستمر، أي أنه يعرف كيف يتوصل الى الحل ولكن التنفيـذ مضجر. أمـا إذا كانت لدينا قائمة من الأرقام الكبيرة ونريـد أن نعرف حـاصل ضـرب كل منهـا مـع أحـد العددين المـذكورين أعـلاه فإن التـوصل الى حـل نهـائي سيكون مضجراً بالتأكيد. هنا سيكون استخدام الحاسبة الالكترونية مفيـداً، بحيث نبـرمجها بخـطوات الحل التفصيليـة، فتنجز الحـاسبـة الالكتـرونيـة المهمات الموكلة إليها بسرعة فائقة ودقة موثوق بنتائجها.

لذا فالبرمجة هي عملية تحليل وتخطيط خطوات حـل مسألـة ومن ثم تشفيـرها بلغـة برمجـة تستطيـع الحـاسبـة الالكتـرونيـة من تنفيـذ أوامـرهـا وعبـاراتها. جـزء من التحدي الـذي يكمن في البرمجـة هو استـطاعـة كـل مبـرمج من تصميم استـراتيجية حـل مسألـة تختلف عن تلك التي يصممهـا آخـر، كلٌ حسب طـريقة تفكيـره وأسلوبه في التحليـل. ولكن تجد معـظم المبـرمجين يتسلسلون في الإجابـة على الأسئلة التاليـة عندمـا يريـدون حل مسألة باستخدام الحاسبة الالكترونية:

أولاً : هـل تم ذكر خصـائص المسـألـة بصـورة واضحـة وعلى نحـو كامل؟ أي هل تم تعريف المسألة من حيث البيـانات المتـوفرة وعمليات المعالجة المطلوب تنفيذها، ومن ثم مـا هي النتائـج المطلوبة.

ثانياً : هل أنا أعرف طريقة لحل تلك المسألة؟

إن إجراءات حل مسألة خطوة بخطوة على نحو تفصيلي ودقيق هي ما يدعى بالخوارزمية. فإذا كان للمبرمج معرفة شخصية بخوارزمية حل فما عليه سوى أن يترجمها الى لغة برمجة ليتصل بالحاسبة الالكترونية التي تحت تصرفه. أما إذا لم تكن لديه طريقة أو إجراء أو خوارزمية للحل فعليه أن يأخذ بنظر الاعتبار السؤال التالي.

ثالثاً : هل أستطيع الحصول على طريقة حل لتلك المسألة من أشخاص آخرين أو بعض الكتب أو المجلات المتخصصة؟

دائماً يوجد حل كلي أو جزئي للمسألة قيد الدراسة. وإذا توفر ذلك فما على المبرمج سوى أن يكيّف طريقة الحل لمعطيات مسألته هو، ومن ثم يترجمها الى لغة برمجة مختارة. أما إذا لم تتوفر خوارزمية حل، فعليه أن يأخذ بنظر الاعتبار السؤال التالي,

رابعاً : كيف سأصمم خوارزمية حل تلك المسألة؟

هنا تكمن العملية الإبداعية عند برمجة حل مسألة. من الطروح الفعالة في تحليل مسألة، هي بتجزئتها الى مهمات أصغر سهلة الفهم. ومن ثم توضع خطوات دقيقة وتفصيلية للتوصل الى حل تلك المهمات الصغيرة. عندها يستطيع المبرمج من ترجمة خوارزمية الحل الى لغة مختارة. توجد أدوات تحليلية لبرمجة الحاسبة الالكترونية ستتعرف عليها من خلال دراستك للفقرات المقبلة من هذا الفصل.

ستـزداد معرفتـك بكيفية تتبـع الأسئلة الأربعة أعـلاه من خلال تتبعك لأفكار هذا الفصل الخاص بالبرمجة .

اختبر نفسك :

1 ـ ما هي البرمجة؟

2 ـ لماذا نستخدم الحاسبة الالكترونية في حل المسائل؟

3 ـ ما هي الخوارزمية؟

4 ـ مـا هي الأسئلة الأربعـة التي يجب أن يعـرف إجـابتهـا الشخص الـذي يريد برمجة حل مسألة معينة؟

5 ـ مـا هي المصادر التي يمكن الـرجوع إليهـا للحصول على طـريقة حـل مسألة معينة؟

3. خطوات حل مسألة باستخدام الحاسبة الالكترونية :

لقد وضع علماءالحاسبات الالكترويننة خطوات عامة يمكن تتبعها للوصول الى برنامج حل مسألة معينة. تدعى هذه الخطوات العامة بعملية البرمجة. مع الأخذ بنظر الاعتبار أن المسألة تصلح للحل باستخدام الحاسبة الالكترونية. أما الخطوات فهي كما يلي :

أولاً : تعرّف المسألة أولاً بتحليل النتائج المطلوبة ومن ثم مراجعة البيانات المتوفرة وأخيراً تحديد خطوات معالجة البيانات المطلوبة.

ثانياً : هل المسألة هي مناسبة للحل بواسطة الحاسبة الالكترونية. حيث تعتبر الحاسبة الالكترونية أداة مثالية فيما إذا تعلق الأمر بمعالجة بيانات من الأنواع التالية :

(أ) إضافة، طرح، ضرب وقسمة البيانات.

(ب) مقارنة البيانات.

(جـ) خزن واسترجاع البيانات.

ثالثاً : تصميم الانسياب المنطقي للحل باستخدام طريقة المخطط الانسيابي أو طريقة شبه الشفرة. وسيتم شرح كل منهما

بفقرات لاحقة من هذا الفصل. إن نتيجة الخطوة هو ما يدعى بالخوارزمية (Algorithm) .

رابعاً : ترجم خطوات خوارزمية الحل الى احدى لغات الحاسبة الالكترونية. إن نتيجة هذه الخطوة هو ما يدعى بالبرنامج (program) . يجب أن يتسم البرنامج الجيد بالصفات التالية:

(أ) يجب أن يكون البرنامج سهل القراءة.

(ب) يجب أن يكون البرنامج موثوق بنتائجه.

(جـ) يجب أن يكون البرنامج تحت جميع الشروط.

(د) يجب أن يكون البرنامج سهل التصحيح.

خامساً : ادخل البرنامج الى الحاسبة الالكترونية.

سادساً : ادرس نتائج البرنامج بعد اختباره بعينة من البيانات المختارة للتأكد من عمل البرنامج تحت جميع الشروط.

نلفت انتباهك، عزيزي القارىء، في هذه المرحلة الى أهمية الوقت عند أساتذة البرمجة. حيث تجدهم يؤكدون على تحديد وقت معين لحل مسألة باستخدام حاسبة الكترونية. ومن ثم تقسيم ذلك الوقت المحدد بين خطوات الحل المذكورة أعلاه وكما يلي:

أولاً : (40٪) للخطوات الثلاثة الأولى والتي تتوج بكتابة خوارزمية الحل.

ثانياً : (20٪) لكتابة البرنامج.

ثالثاً : (40٪) للخطوتين الأخيرتين. أي لتغذية البرنامج واختباره

وتدقيق النتائج للوصول الى البرنامج الجيد الذي يعطي حلاً مناسباً للمسألة قيد الدراسة.

سنكرس ما تبقى من هذا الفصل لارشاد القارىء الى كيفية كتابة خوارزمية ناجحة لحل مسألة معينة. أما تعلم كتابة برنامج بلغة بيسك فسيتم عن طريق متابعة دراسة فصول هذا الكتاب، ما عدا الفصل الرابع الذي هدفه إرشاد القارىء الى كيفية التعامل مع جهاز الحاسبة الالكترونية لتنفيذ برنامج بلغة بيسك.

4. الخوارزمية :

عند تحدثنا عن السؤال الثاني الذي يستحسن أن يجيب عليه كل شخص يريد برمجة حاسبة الكترونية لحل مسألة، فإننا قد ذكرنا الخوارزمية على أنها مجموعة من خطوات دقيقة وتفصيلية لحل مسألة تأخذ بنظر الاعتبار كل الشروط والاحتمالات. يمكن ذكر خصائص يجب أن تنفرد بها الخوارزمية منها:

(أ) تتألف الخوارزمية من خطوات معدودة.
(ب) تكون خطوات الخوارزمية دقيقة.
(جـ) تكون خطوات الخوارزمية واضحة.
(د) يجب أن تكون الخوارزمية عامة في صيانتها.

يعتقد الكثير من أساتذة برمجة الحاسبة الالكترونية، أن كتابة الخوارزمية هي من أهم مراحل كتابة البرنامج للحاسبة الالكترونية فكما أن بناء البيت يتطلب وجود خريطة تبين مساحة وعدد الغرف وحدودها ومواقعها وغير ذلك من التفاصيل التي يستعان بها في بناء البيت، كذلك فإن الخوارزمية تعتبر كالخريطة التي يستعين بها المبرمج عند كتابة برنامج حل مسألة.

الآن، يحق لنا أن نتسائل ما الذي ستوفره لنا الخوارزمية بحيث نعطيها كل هذه الأهمية والخصوصية؟ إن أهمية خوارزمية الحل نابعة من تجهيزها لعدة أمور حيوية لكتابة برنامج ناجح، ومن هذه الأمور:

أولاً : تعطي الخوارزمية صورة واضحة عن كيفية عمل البرنامج بحيث تريدها أن ترينا ماذا يفعل البرنامج؟ وكيف يعالج الأنواع المختلفة من البيانات؟

ثانياً : وصف يحتوي على مجموعة من الخطوات التي نستطيع ترجمتها بسهولة الى احدى لغات برمجة الحاسبة الالكترونية.

ثالثاً : طريقة لتجزئة البرامج المعقدة الى أجزا ء يمكن إدارتها أجزاء وتدقيقها ومن ثم دمجها لتكون البرنامج الكلي.

رابعاً : طريقة للتعرف على الأخطاء بحيث تستطيع تصحيحها قبل كتابة البرنامج أو بعد تنفيذ البرنامج بالنسبة للأخطاء المنطقية.

خامساً : وسيلة لشرح وتوثيق حل المسألة بصورة نظامية للآخرين من زملاء ومبرمجين.

سادساً : تساعد الخوارزمية على تجنب نسيان خطوات حل مهمة.

وبذلك تساعد الخوارزمية على تنظيم أفكار المبرمج حتى قبل أن يجلس الى الحاسبة الالكترونية، هنالك عدة أدوات تحليلية يمكن بواسطتها كتابة خوارزمية لحل مسألة باستخدام الحاسبة الالكترونية، ومن هذه الأدوات يمكن أن نذكر ثلاثة ذات أهمية وفعالية:

(أ) المخطط الانسيابي (Flow chart)

(ب) شبه الشفرة (pseudo Code)

(جـ) البرمجة الهيكلية (structured programming)

سنشرح كل واحدة منهما بالتفصيل في الفقرات المقبلة.

اختبر نفسك :

1 ـ اذكر خطوات حل مسألة معينة باستخدام الحاسبة الالكترونية؟

2 ـ ما الذي تستطيع الحاسبة الالكترونية عمله؟

3 ـ ما هي صفات البرنامج الجيد؟

4 ـ كيف تم تقسيم الوقت المحدد لحل مسألة باستخدام الحاسبة الالكترونية؟

5 ـ ما هي الخوارزمية؟ وما هي خصائصها؟ وما الذي توفره؟ اذكر ثلاثة أدوات تحليلية يمكن بواسطتها كتابة خوارزمية لحل مسألة لكي يتم برمجتها بلغة حاسبة الكترونية.

5. المخطط الانسيابي :

من المهم أن يتعرف القارىء على المخطط الانسيابي كأداة لكتابة الخوارزمية قبل تعلمه للغة برمجة البيسك لكي :

أولاً : يبدأ القارىء في تنمية الانضباط والجهد العقلي الذي تحتاجه عملية البرمجة.

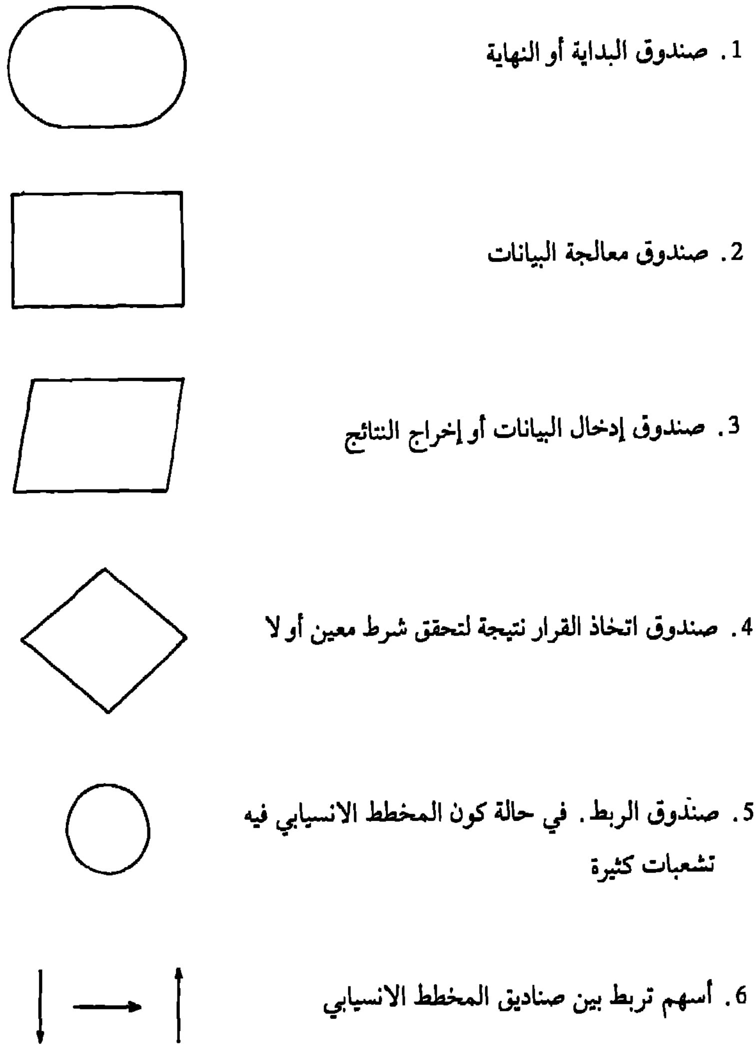
ثانياً : يوجد وصف مختصر للحل بدون اللجوء إلى الشرح المطول لاجراءات الحل.

ثالثاً : يصبح الحل مجسداً أمام القارىء بصورة إجراءات.

ابتكر علماء برمجة الحاسبات الالكترونية طريقة وصف تصويرية لخطة تنفيذ خطوات حل مسألة معينة تمهيداً لبرمجتها بلغة حاسبة الكترونية مناسبة. تدعى هذه الطريقة بالمخطط الانسيابي. يتكون المخطط الانسيابي من صناديق ذات أشكال هندسية ترمز كل منها لاحدى وظائف معالجة البيانات في الحاسة الالكترونية. ترتبط تلك الأشكال الهندسية مع بعضها بخطوط ذات أسهم تدل على اتجاه متابعة الحل منطقياً. يتم كتابة خطوات الحل (أي التعليمات) داخل الأشكال الهندسية.

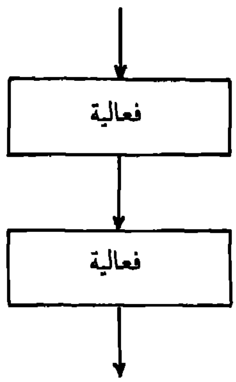
يتميز استخدام الحاسبة الالكترونية في حل المسائل بمراحل ثلاثة هي تغذية البيانات وتعليمات معالجتها ومن ثم إظهار نتائج تلك المعالجة. لذلك عند كتابة خوارزمية حل باستخدام المخطط الانسيابي فستجد هنالك شكل متوازي أضلاع يكتب في داخله تعليمات إدخال البيانات أو إخراج النتائج كما يوجد شكل مستطيل يكتب في داخله تعليمات معالجة البيانات. أما إذا أردت التحقق من شرط لاتخاذ قرار معين فقد ابتكر رمز معيني يكتب في داخله الشرط، ويخرج منه سهمان يبينان الإجراءات التي ستنفذ في حالة تحقق الشرط أو عدم تحققه. أخيراً ابتكر رمز بداية المخطط الانسيابي أو انتهائه ويكون بيضوياً ليعلن عن الابتداء أو النهاية. أنظر شكل (1 . 1) لترى الأشكال الهندسية لرموز المخطط الانسيابي المختلفة.

بعد تمرسك على كتابة الخوارزمية بواسطة المخطط الانسيابي فإنك ستتأكد أن المخطط الانسيابي لحل أي مسألة ما هو إلا تشكيلة من تركيبات لأنماط منطقية. ويوجد أربعة أنواع من تركيبات الأنماط المنطقية:



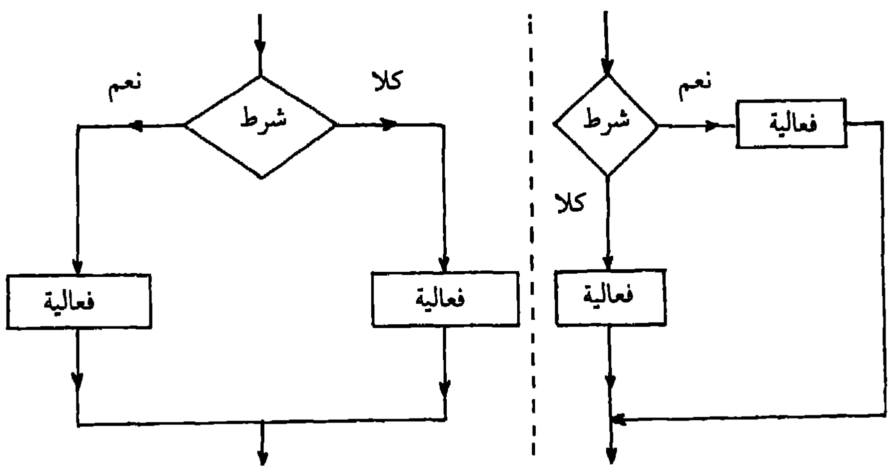
شكل (1.1) رموز المخطط الانسيابي

أولاً : تركيبة التسلسل البسيط: وهي الاستمرار في تنفيذ الفعاليات المعالجة الواحدة بعد الأخرى بصورة متتالية كما في الشكل (1 . 2) .



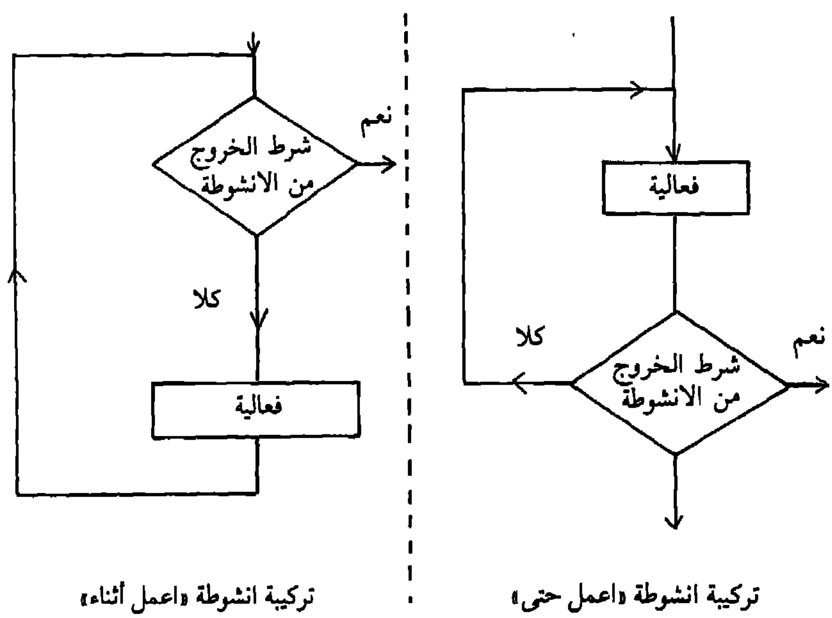
شكل (1 . 2) تركيبة التسلسل البسيط

ثانياً : تركيبة الاختيار: أو ما يعرف بالانتقال الشرطي والذي يعتمد على اختبار تحقق شرط معين. والجواب يكون إما (نعم) عند تحقق الشرط أو (كلا) عند عدم تحققه ونتيجة لذلك سيتم اتباع احدى اتجاهين كما في شكل (1 . 3) .



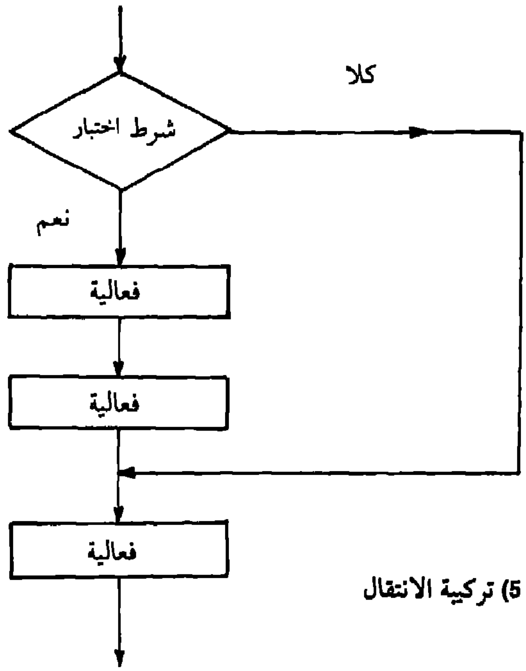
شكل (1 . 3) تركيبة الاختيار مرسوم بهيئتين مختلفتين

ثالثاً : تركيبة الأنشـوطة: أمـا ما تـدعى بتركيبـة التكرار. وهي تكـرار تنفيذ الفعاليات معالجة البيانات في البرمجة. ولن يستمر هـذا التكـرار الى مـا لانهـايـة بـل يجب وضـع شـرط للخـروج من الانشوطة. يكون موقع الشرط إما في بداية الانشوطـة وعندهـا تدعى بتركيبة انشوطة «اعمل أثناء» أو يكون الشـرط عند نهـاية الانشـوطة فتـدعى بتركيبـة انشـوطـة «اعمـل حتى». كمـا في شكل (1 . 4) .



شكل (1 . 4) تركيبة الانشوطة

رابعاً : تركيبة الانتقال: وهي التي تعبر عن منـطق القفز لتنفيـذ فعاليـة معالجة بيانات مناسبة استنـاداً الى شرط معين. كمـا في شكل (5 . 1) .



شكل (5 . 1) تركيبة الانتقال

بعد أن تعرفنا على مفهوم المخطط الانسايبي لكتـابة خـوارزمية حـل مسـألـة، علينـا الآن استخـدامـه في مثـال بسيط من حيـاتنـا اليـوميـة قبـل استخدامه في التخطيط لبرمجة الحاسبة الالكترونية المثال أدنـاه مقتبس من ورقة امتحانية (East Anglian Examination Board) .

مثـال :

عنـد عودتـك من المدرسـة أو الجامعـة أو الوظيفـة الى بيتك فـإنـك

ستخطط لقضاء امسيتك. والمخطط الانسيابي أدناه يعطيك فكرة بسيطة لما تنوي عمله. أعد رسم هذا المخطط الانسيابي مؤشراً عليه التعليمات المفقودة والتي قد جمعت بجانب المخطط الانسيابي.

تعليمات:

((أ) اقرأ الجريدة.

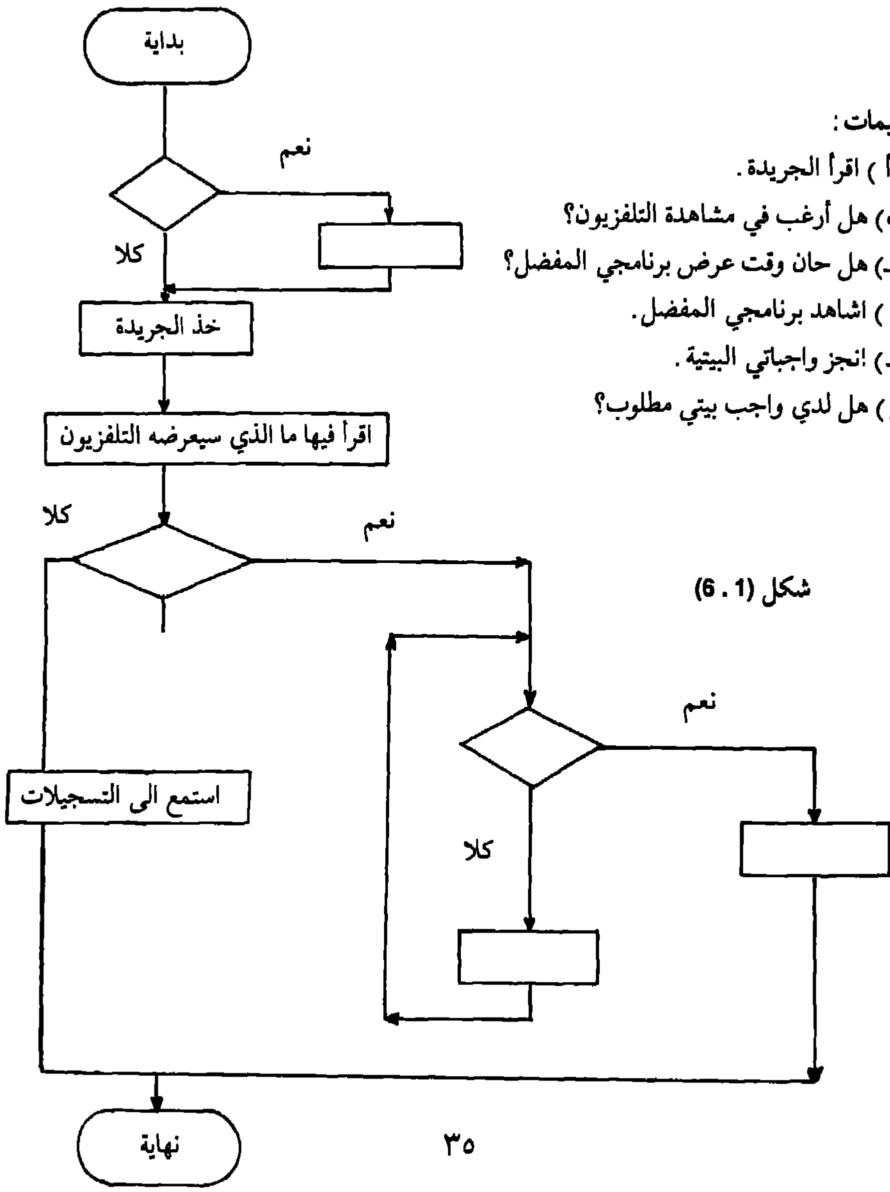
(ب) هل أرغب في مشاهدة التلفزيون؟

(جـ) هل حان وقت عرض برنامجي المفضل؟

(د) اشاهد برنامجي المفضل.

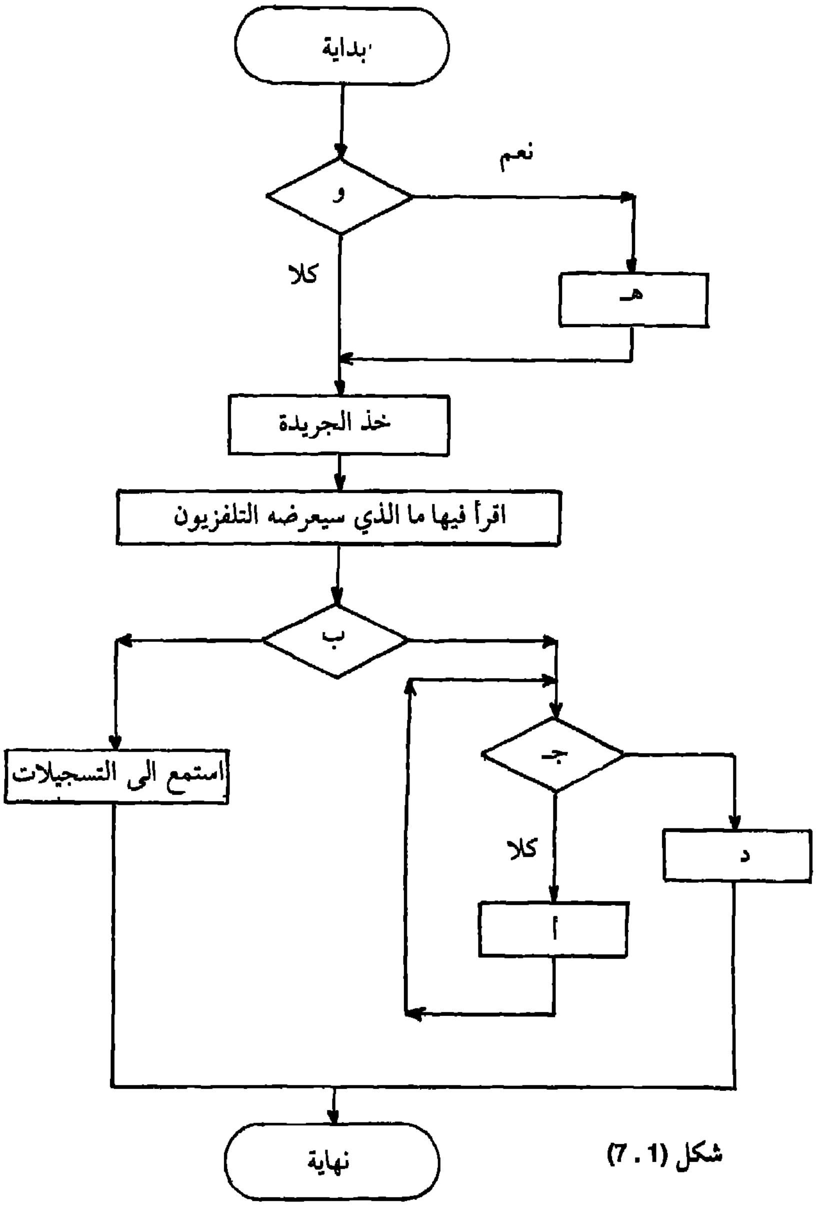
(هـ) انجز واجباتي البيتية.

(و) هل لدي واجب بيتي مطلوب؟



شكل (6 . 1)

الحـــل : سيكون الحل كما في الشكل (1 . 7) .



شكل (1 . 7)

بعـد تفحصك للمخـطط الانسيابي في شكـل (1 . 7) ، لا بد وأنـك قد لاحظت ما يلي :

أولاً : يكـون الاتجاه العـام لسريـان تنفيـذ العمليـات من الأعلى الى الأسفــل وإذا اتجـه الى أعلى فلتكملة انشــوطــة (loop) أي لتكرار انجاز بعض العمليات .

ثانياً : في الرمز المستطيل الخـاص بالمعـالجة ، يـدخل خط سـريان واحد ويخرج كذلك خط واحد فقط . أما في رمز القرار المعيني فيـدخـل خط سـريـان واحـد ويخـرج منـه خـطان يكتب على احدهما كلمـة (نعم) وعلى الأخرى (كـلا) ، تعبيراً عن اتجـاه التنفيذ عند تحقق الشرط الموجود داخل الـرمز أو عـدم تحققه على التعاقب .

ثالثاً : مع الرمز الطرفي يـوجد خط سـريان واحـد ، يكون خـارج من الـرمـز الـطرفي عنـد البـدايـة ، وداخــلاً الى الـرمــز الـطرفي المستخدم عند نهاية المخطط الانسيابي .

رابعاً : تكوّن المخطط الانسيابي من تركيبات لأنماط منطقية .

خامساً : لاحظ كيفيـة ترابط رمـوز المخـطط الانسيـابي لتصـويـر فكـرة الحل .

اختبر نفسك :

1 ـ ما هو المخطط الانسيابي؟

2 ـ لماذا يجب دراسة المخطط الانسيابي قبل تعلم البرمجة بلغة بيسك؟

3 ـ ما هي رموز المخطط الانسيابي؟ وكيف تستخدم لكتابة الخوارزمية؟

4 ـ ما هي تركيبات الأنماط المنطقية؟ اذكر أنواعها.

5 ـ ارسم مخططاً انسيابياً لايجاد معدل درجاتك في درس الرياضيات. علماً أن لديك عشرة درجات.

6. شبه الشفرة :

يفضل بعض المبرمجين كتابة خوارزمية حل مسألة باستخدام كلمات اللغة البشرية التي يجيدونها (كالعربية أو الانكليزية أو الفرنسية...). حيث تجد تعليمات خوارزمية الحل عبارة عن جمل لغوية بدلاً من رموز لأشكال هندسية مرتبطة مع بعضها بخطوط انسياب ذات أسهم كما في طريقة المخطط الانسيابي. والفائدة من شبه الشفرة هي تذكير المبرمج بمنطق الحل وخطواته التفصيلية.

لنعيد كتابة خوارزمية قضاء أمسية، والمصورة على شكل مخطط انسيابي في القفرة (5) من هذا الفصل، باستخدام طريقة شبه الشفرة.

خوارزمية قضاء أمسية.

1 ـ هل لدي واجب بيتي مطلوب؟ إذا كان الجواب بنعم نفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (3) .

2 ـ انجز واجباتي البيتية.

3 ـ خذ الجريدة.

4 ـ اقرأ فيها ما الذي سيعرضه التلفزيون.

5 ـ هل أرغب بمشاهدة التلفزيون؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (11) .

6 ـ هل حان وقت عرض البرنامج المفضل؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (12) .

7 ـ اشاهد برنامجي المفضل.

8 ـ اذهب الى الخطوة رقم (12) .

9 ـ اقرأ الجريدة.

10 ـ اذهب الى الخطوة رقم (16) .

11 ـ استمع الى التسجيلات.

12 ـ توقف.

نهاية الخوارزمية.

بعد كتابة خوارزمية قضاء أمسية بأسلوبي المخطط الانسيابي وشبه الشفرة، فإن القارىء قد عرف الفرق بينهما ولكنه ربما يسأل نفسه؛ متى استخدم احد الأسلوبين وأفضله على الآخر؟ يعتبر أسلوب المخطط الانسيابي مفيد في جميع الحالات، ولكن يفضل استخدام شبه الشفرة عند كتابة الخوارزمية في الحالات التالية:

أولاً : إذا كان هنالك قرارات كثيرة والذي يحتم تقاطع خطوط القرار وبالتالي تعقد المخطط الانسيابي بحيث يصبح غير مفهوم.

ثانياً : عدم كفاية التفاصيل الموجودة داخل الأشكال الهندسية مما يتطلب وصفاً لغوياً عند كتابة الخوارزمة .

7. البرمجة الهيكلية :

عندما تكون المسألة معقدة جداً فإن أسهل طريقة لكتابة خوارزمية الحل هي تجزئة المسألة الى مهمات صغيرة بحيث تؤدي الخوارزمية الوظيفة الرئيسية في حل المسألة. يدعى هذا الأسلوب في برمجة الحاسبة الالكترونية بالبرمجة الهيكلية (structured programming) ـ تنصح البرمجة الهيكلية باتباع الخطوط التحليلية التالية:

أولاً : تجزئة البرنامج الرئيسي الى وحدات مستقلة يدعى كل منها بالبرنامج الفرعي(Subroutine). وهو برنامج صغير يعمل ضمن برنامج رئيسي. يتميز كل برنامج فرعي :

(أ) يؤدي البرنامج الفرعي مهمة صغيرة واحدة.

(ب) يكون البرنامج الفرعي بسيطاً قليل الخطوات.

(جـ) يكون للبرنامج الفرعي نقطة دخول واحدة ونقطة خروج واحدة.

ثانياً : تشجع استخدام تركيبات الأنماط المنطقية التالية :

(أ) تركيبة التسلسل البسيط.

(ب) تركيبة الاختيار.

(جـ) تركيبة الانشوطة.

ولا تشجع على استخدام تـركيبة الانتقـال لأنها تغيـر باستمـرار مجـرى تنفيذ البـرنـامـج وبـالتـالي الى صعـوبـات في اختيـار البرنامج وتصحيح أخطائه ومتابعته.

يمكن استخـدام المخـطط الانسيـابي أو شبـه الشفـرة عنـد كتـابـة الخوارزمية بـأسلوب البرمجـة الهيكلية. والآن سنعـود الى خوارزميـة قضاء أمسية وسنحللها باتباع اسلوب البرمجة الهيكليـة. سنكتب الخوارزميـة أولاً بشبه الشفرة كما يلي:

خوارزمية قضاء أمسية.
1 ـ إنجاز الواجب البيتي.
2 ـ قراءة الجريدة.
3 ـ مشاهدة التلفزيون.
نهاية الخوارزمية.

تعتبر الخطوات الثلاثة أعـلاه هي البرنـامج الـرئيسي مكتوبـاً بشبه الشفـرة. ويمكن كتابة كل خطوة منه كبرنامج فرعي منفصل كما يلي:

إنجاز الواجب البيتي.
1 ـ هل لدي واجب بيتي؟ إذا كـان الجواب بنعم فنفـذ الخطوة التـالية. أمـا إذا كان الجـواب بكلا فـارجع الى البـرنـامـج الرئيسي.
2 ـ انجز واجباتي البيتية.
عودة.

تعني كلمة «عودة» (Return)، أن ارجع الى الخطوة التالية في البرنامج الرئيسي. يؤدي هذا البرنامج الفرعي مهمة واحدة هي إنجاز الواجب البيتي. يمكن كتابة الخطوة رقم (2) من البرنامج الرئيسي كبرنامج فرعي كما في شبه الشفرة التالية:

قراءة الجريدة.
1 ـ احصل على جريدة.
2 ـ اقرأ فيها ما الذي سيعرضه التلفزيون.
عودة.

يؤدي هذا البرنامج الفرعي مهمة واحدة هي قراءة الجريدة. وهو كما ترى مختصر وله نقطة دخول واحدة ونقطة خروج واحدة أيضاً. أما الخطوة الثالثة من البرنامج الرئيسي فهي برنامج فرعي أيضاً يمكن كتابته بشبه الشفرة التالية:

مشاهدة التلفزيون
1 ـ هل أرغب بمشاهدة التلفزيون؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (7).
2 ـ هل حان وقت عرض برنامجي المفضل؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (6).
3 ـ اشاهد برنامجي المفضل.

4 ـ توقف.

5 ـ اقرأ الجريدة.

6 ـ اذهب الى الخطوة رقم (2).

7 ـ استمع الى التسجيلات.

8 ـ توقف.

عودة.

بعد أن تم كتابة خوارزمية قضاء أمسية بأسلوب البرمجة الهيكلبة أصبحت عبارة عن برنامج رئيسي وثلاثة برامج فرعية. مما تقدم لا بد أنك قد لاحظت الأمور التالية:

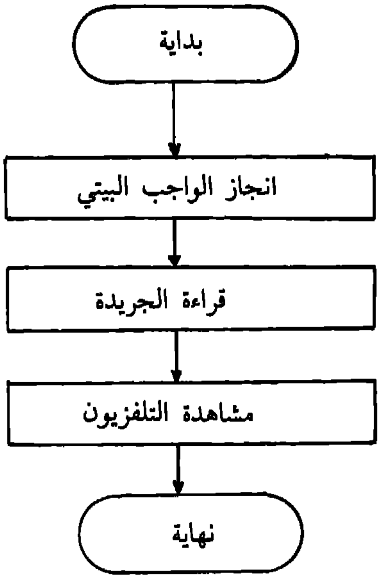
أولاً : تم تقليل تعقيد الخوارزمية الرئيسية للحل.

ثانياً : تقليل الجهد المبذول، من قبل المبرمج والآخرين في متابعة تحويل الخوارزمية الى برنامج حاسبة الكترونية.

ثالثاً : قد اتبع أسلوب التدرج من المهمة الرئيسية الى مهمات فرعية أي التدرج من الأعلى الى أسفل.

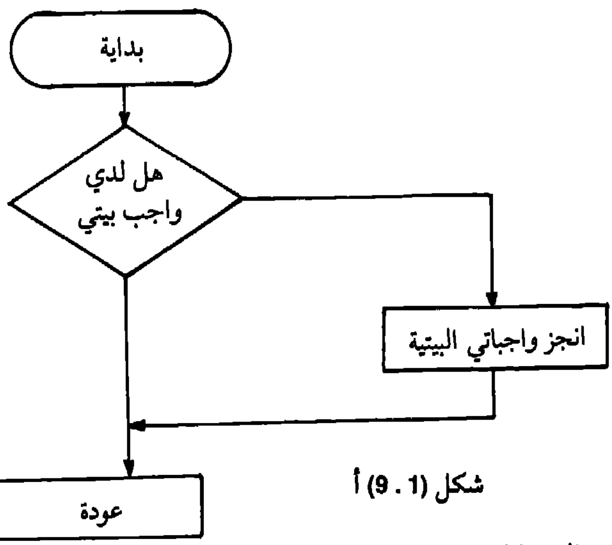
رابعاً : من الأمور الثلاثة أعلاه فإن وقت اختبار البرنامج سيقل حتماً.

يمكننا الآن كتابة خوارزمية قضاء أمسية ذات البرمجة الهيكلية باستخدام المخطط الانسيابي، وكما يلي:



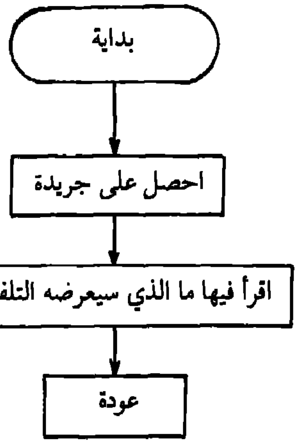
شكل (1 . 8) المخطط الانسيابي لخوارزمية قضاء أمسية

ويمكن تصوير البرامج الفرعية للخوارزمية اعلاه بمخططات انسيابية كما في الشكـــل التـالي:

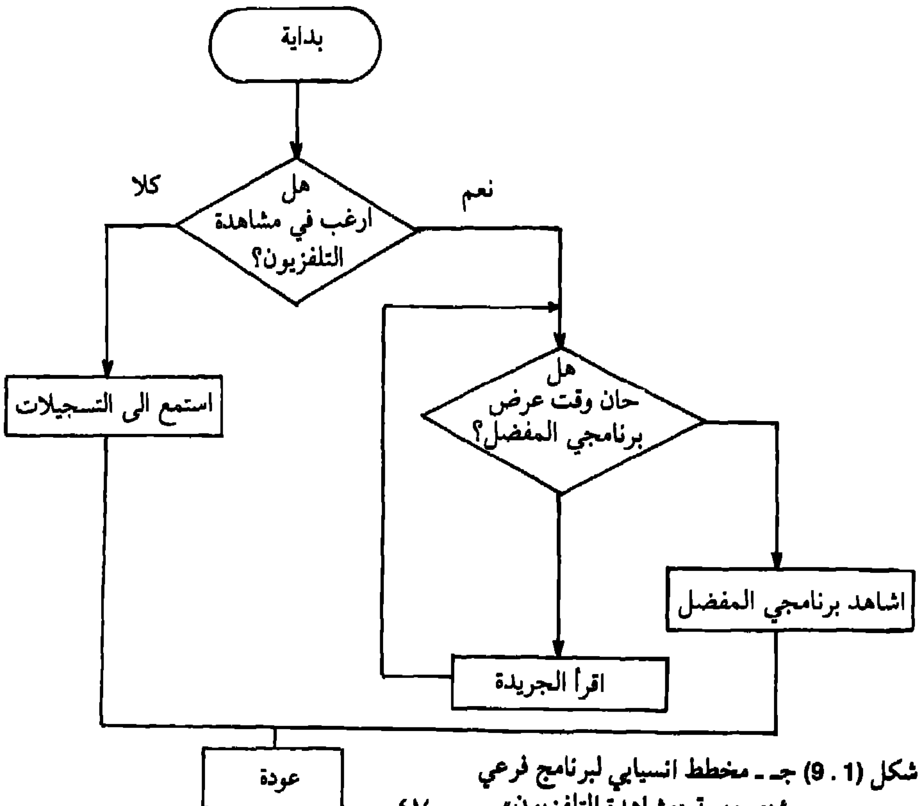


شكل (1 . 9) أ

ـ المخطط الانسيابي للبرنامج الفرعي الذي «ينجز الواجب البيتي»



شكل (1 . 9) ب ـ المخطط الانسيابي للبرنامج الفرعي «لقراءة جريدة»



شكل (1 . 9) جـ ـ مخطط انسيابي لبرنامج فرعي
يؤدي مهمة «مشاهدة التلفزيون»

اختبر نفسك :

1 ـ بماذا تفضل أن تكتب الخـوارزمية بـطريقة المخـطط الانسيابي أم بشبـه الشفرة؟ هل تعرف السبب؟

2 ـ ما هي شبه الشفرة؟

3 ـ ما هي البرمجة الهيكلية؟ ومتى تستخدم؟

4 ـ بماذا تنصحك البرمجة الهيكلية عند كتابة خوارزمية حل مسألة؟

5 ـ ما الكلمة التي تستخدم لتربط برنامج فرعي مع البرنامج الرئيسي؟

8. مسائل محلولة :

وضعنا في هذه الفقرة ستة مسائل لنبين للقارىء الكريم «كيف يبرمج حاسبة الكترونية». تدور المسائل الستة حول نفس الفكرة تقريباً ولكن تم التخطيط لكتابة خوارزمية برنامجها باتباع طرق وأساليب برمجة متنوعة فادرس وتعلم.

مسألة (1) : في نهاية كل شهر يقبض العاملون في مصنع للحاسبات الالكترونية أجوراً عن الساعات الإضافية. ويكون المبلغ عبارة عن عدد الساعات الإضافية مضروباً في أجر كل ساعة. فإذا كان عدد العاملين هو مائة، ارسم مخططاً انسيابياً للخطوات التي سيتبعها المحاسب لكي يعطي استحقاق كل مشتغل عن عمله في الساعات الإضافية.

الحـــل : يجب ملاحظة ما يلي من نص المسألة أعلاه.

أولاً : إن معطيات المسألة هي عدد الساعات الإضافية، وأجر كل ساعة، لذلك سنستخدم رمز الإدخال المتوازي الأضلاع ونكتب فيه تلك المعطيات.

ثانياً : يتم حساب المبلغ الكلي من الصيغة الرياضية التالية:

المبلغ الكلي = عدد الساعات الإضافية × أجر كل ساعة.
لذا يجب استخدام رمز المعالجة المستطيل ويكتب فيه الصيغة الرياضية أعلاه.

ثالثاً : لكون عدد العاملين هو مائة فسنستخدم عدّاداً (counter) وانشوطة (loop) لتكرار العملية الحسابية المذكورة في (2) مائة مرة على البيانات المتوفرة.

رابعاً : النتيجة المطلوبة هي المبلغ الذي يجب دفعه لكل مشتغل في نهاية كل شهر. سنستخدم رمز الاخراج المتوازي الأضلاع ونكتب فيه ما المطلوب إخراجه كنتيجة.

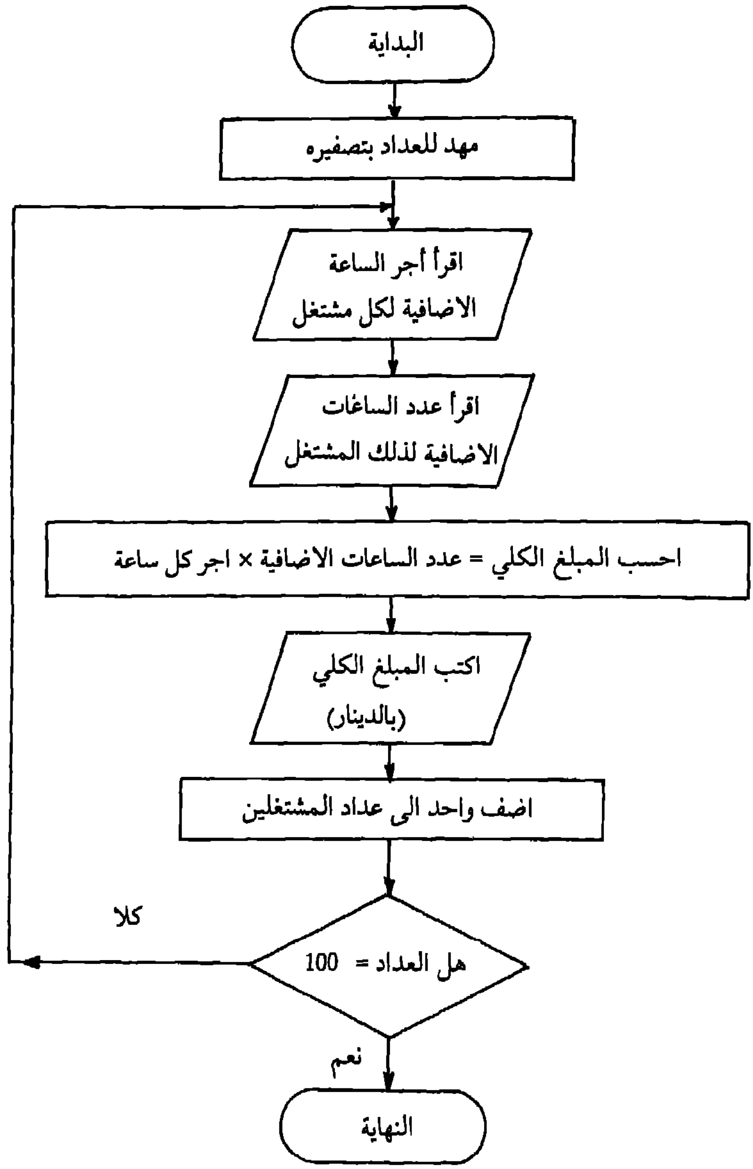
وبذلك ستبدو خوارزمية حل المسألة (1) بالمخطط الانسيابي التالي:

لاحظ ما يلي من المخطط الانسيابي:

أولاً : استخدم السؤال «هل العداد = 100؟» كوسيلة للخروج من الانشوطة وإلا أصبحت الانشوطة غير منتهية أي ستتكرر الى ما لانهاية من المرات.

ثانياً : استخدم العداد كوسيلة لحساب عدد المشتغلين. يجب التمهيد للعداد بتصفيره.

ثالثاً : قد وضعنا أجر الساعة الإضافية لكل مشتغل ضمن الانشوطة لافتراضنا أن هنالك بعض التفاوت في اجر الساعة الإضافية بين المشتغلين.



شكل (1 . 10) المخطط الانسيابي لخوارزمية حل المسألة (1)

رابعاً : يتكون المخطط الانسيابي في شكل (1 . 10) من تركيبة التسلسل البسيط وتركيبة انشوطة «اعمل حتى».

مسألة (2) : اكتب خوارزمية حل مسألة دفع أجور الساعات الإضافية للعاملين في مصنع للحاسبات الالكترونية المذكور في مسألة (1)، باستخدام شبه الشفرة.

الحل : يمكننا كتابة شبه الشفرة لخوارزمية الحل وذلك بإدخال المعطيات، ثم معالجتها رياضياً للتوصل الى المبلغ الكلي للساعات الإضافية لكل مشتغل، عندها نخرج النتائج. سنستخدم عداداً وانشوطة لحل المسألة وكما في شبه الشفرة التالية:

خوارزمية أجور الساعات الإضافية

1 ـ مهّد للعداد بتصفيره.

2 ـ اقرأ أجر الساعة الإضافية لهذا المشتغل.

3 ـ اقرأ عدد الساعات الاضافية لهذا المشتغل.

4 ـ احسب: المبلغ الكلي = عدد الساعات الاضافية × اجر كل ساعة. .

5 ـ اخرج المبلغ الكلي (بالدينار).

6 ـ أضف واحد الى عداد المشتغلين.

7 ـ هل العداد = 100؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (2).

8 ـ توقف.

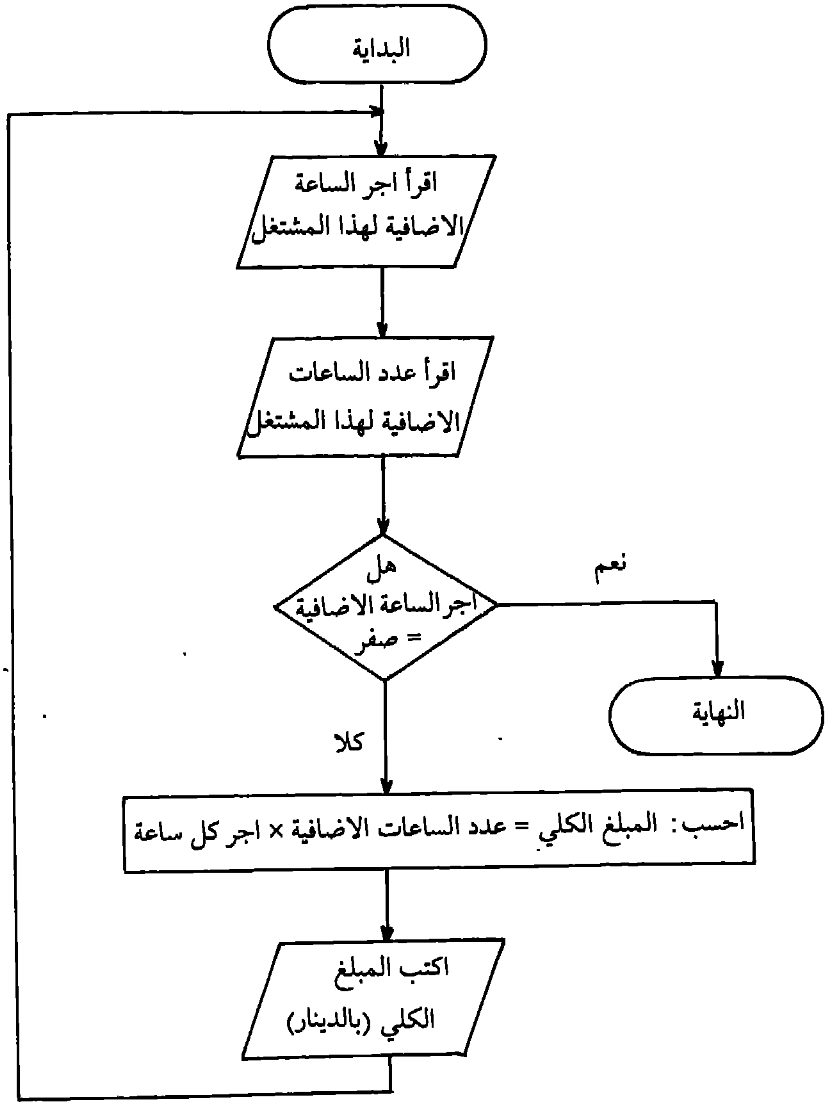
نهاية الخوارزمية.

بعد كتابة شبه الشفرة لخوارزمية حل المسألة (2) ، فقد لاحظت في الخطوة رقم (7) عندما قلنا «اذهب الى الخطوة رقم (2)»، فإن ذلك يعني الدخول في انشوطة لتكرار تنفيذ الخطوات من (2) الى (7) لحين تحقق شرط الخروج من الانشوطة (أي عندما يساوي العداد مائة).

مسألة (3) : إذا غيرنا جزء من نص المسألة (1) وجعلنا عدد العاملين غير معروف، نتيجة لتوظيف أو تسريح العديد من المهندسين والفنيين والاداريين في مصنع الحاسبات الالكترونية. كيف ستصبح خوارزمية أجور الساعات الإضافية التي يجب اتباعها لإعطاء كل مشتغل في المعمل استحقاقه من جراء اشتغاله ساعات إضافية. استخدم المخطط الانسيابي في كتابة الخوارزمية.

الحل : في المسألة (1) والمسألة (2) تم السيطرة على تكرار تنفيذ الانشوطة باستخدام العداد بكون عدد العاملين معلوم. أما في مسألتنا هذه فإن عدد العاملين غير معروف مقدماً لذا يمكننا ابتكار فكرة وضع قيمة عددية معينة في نهاية قائمة أجر الساعة الإضافية لكل مشتغل. لتكن تلك القيمة هي صفراً.

قبل البدء برسم المخطط الانسيابي لاحظ المدخلات ونوع المعالجة التي ستتم على البيانات المدخلة ومن ثم ما هي النتائج المطلوبة. سيبدو المخطط الانسيابي لخوارزمية أجور الساعات الاضافية كما يلي:



شكل (1 . 11) المخطط الانسيابي لخوارزمية حل مسألة (3)

لاحظ الأمور التالية من المخطط الانسيابي في شكل (1 . 11) :

أولاً : استخدمنا السؤال «هل أجر الساعة الإضافية = صفر؟ للسيطرة على تكرار تنفيذ الأنشوطة وللخروج منها عند تحقق الشرط.

ثانياً : استخدمنا الرمز المستطيل لمعالجة الدخلات.
استخدمنا رمز متوازي الأضلاع لإدخال المعطيات وإخراج النتائج. واستخدمنا الرمز المعيني لاختبار شرط معين. وربطنا الرموز المختلفة بخطوط إنسياب لها أسهم يدل اتجاهها على منطق تنفيذ الخوارزمية.

ثالثاً : يتكون المخطط الانسيابي في شكل (11.1) من تركيبة نمط منطقي واحد هو تركيبة أنشوطة «أعمل أثناء».

مسألة (4) : أعد كتابة خوارزمية أجور الساعات الاضافية للمسألة (3) باستخدام طريقة شبه الشفرة، وقارنها مع شبه شفرة المسألة (2) .

الحل : عند كتابة خوارزمية حل لمسألة يجب أن تكون الخوارزمية محدودة الخطوات ودقيقة وواضحة وعامة في صياغتها، كما تم ذكره في الفقرة رقم (4) من هذا الفصل. ويجب أن نعرف من نص المسألة نفسها البيانات المتوفرة وعمليات المعالجة المطلوب تنفيذها على تلك البيانات ومن ثم ما هي النتائج المطلوبة. وكل تلك الأمور قد تم ذكرها في نص المسألة. لذا سنكتب خوارزمية أجور الساعات الاضافية بشبه الشفرة، وكما يلي :

خوارزمية أجور الساعات الإضافية

1 ـ اقرأ أجر الساعة الاضافية لهذا المشتغل.

2 ـ اقرأ عدد الساعات الاضافية لهذا المشتغل.

3 ـ هل أجر الساعة الاضافية = صفر؟ إذا كان الجواب بنعم فانهي تنفيذ خطوات الحل. وإذا كان الجواب بكلا فنفذ الخطوة التالية.

4 ـ احسب: المبلغ الكلي = عدد الساعات الاضافية × أجر كل ساعة.

5 ـ اكتب المبلغ الكلي (بالدينار).

6 ـ اذهب الى الخطوة رقم (1) .

نهاية الخوارزمية.

لاحظ أن الخطوتين (1) و (2) هي لإدخال البيانات.

والخطوات (3) و(4)و(6) هي لمعالجة البيانات.

والخطوة (5) هي لاخراج النتائج.

مسألة (5) : نريد حساب الراتب الكلي (بالدينار) الذي يستحقه كل مشتغل في مصنع للحاسبات الالكترونية. علماً أن عدد المشتغلين هو (50) ما بين فني ومهندس وإداري وأن الراتب الكلي = الراتب الشهري + اجور الساعات الاضافية.

البيانات المتوفرة عن كل مشتغل هي:

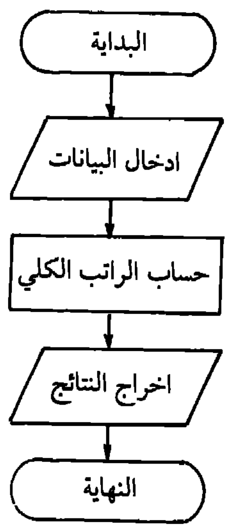
راتبه الشهري.

اجرة كل ساعة إضافية.
عدد الساعات الإضافية.

استخدم أسلوب البرمجة الهيكلية في كتابة خوارزمية حل المسألة أعلاه. واكتب الخوارزمية بمخطط انسيابي.

الحل : ينصح أسلوب البرمجة الهيكلية بتجزئة المسألة الى مهمات صغيرة. يتم التخطيط لبرمجة حل تلك المهمات بوحدات صغيرة تدعى بالبرامج الفرعية. وبذلك تتكون خوارزمية حل مسألة باستخدام البرمجة الهيكلية من برنامج رئيسي وعدة برامج فرعية.

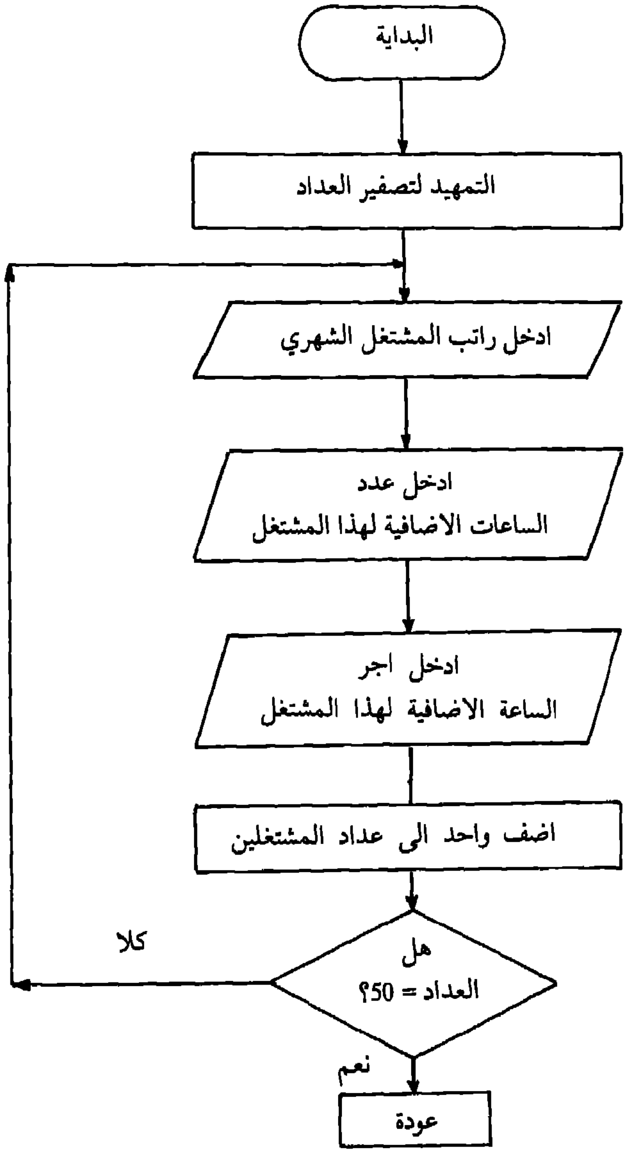
يمكن كتابة خوارزمية حساب الراتب الكلي بالمخطط الانسيابي التالي :



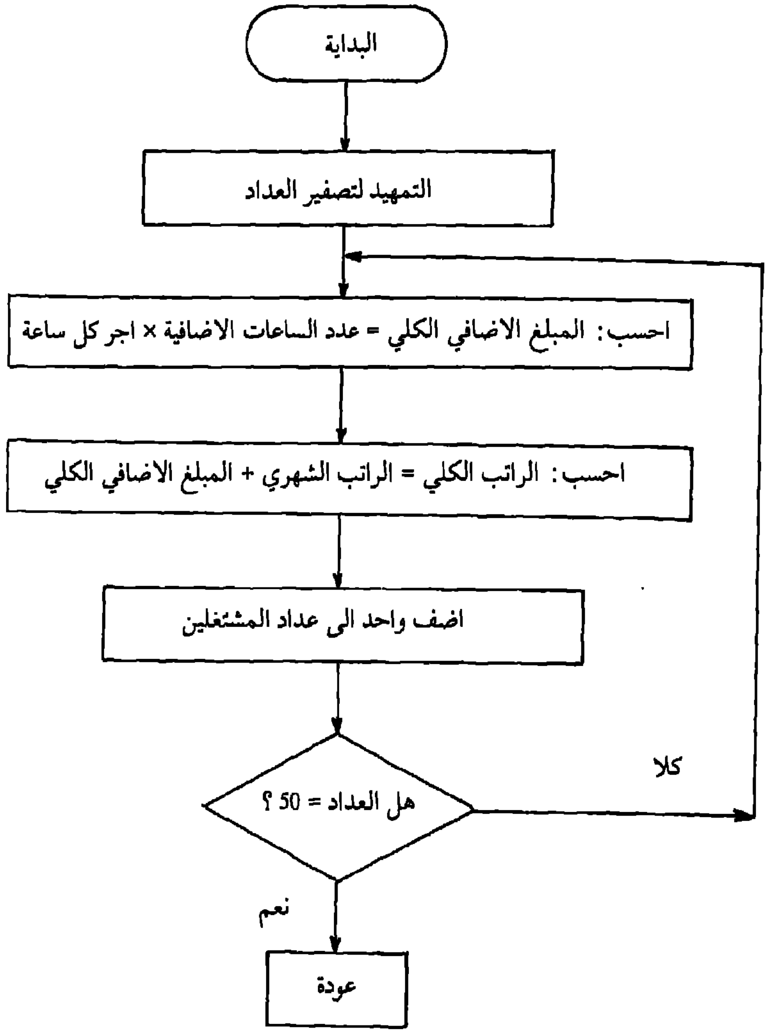
شكل (1 . 12) المخطط الانسيابي لخوارزمية البرنامج الرئيسي

من ملاحظتك لخوارزمية البرنامج الرئيسي في شكل (1 . 12) فإنك

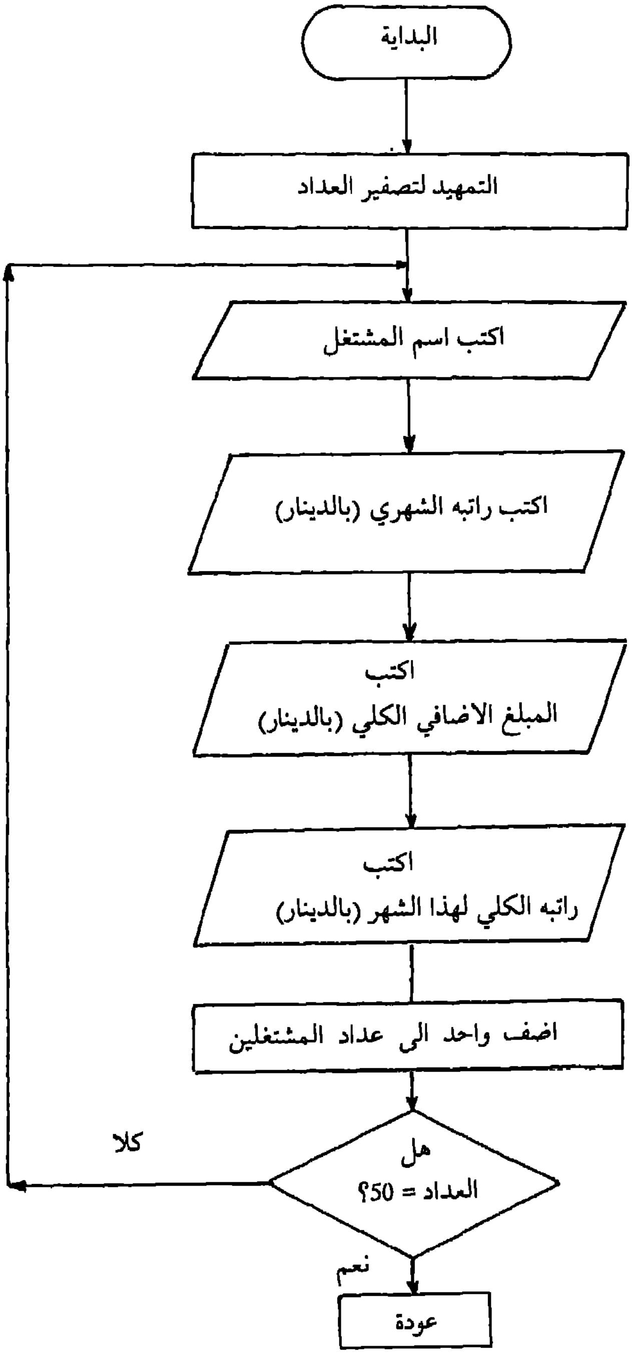
تجـدهـا تتكـون من ثـلاثـة مهمـات . حيث يمكن تحـويـل كـل من تلك المهمات الثلاثة الى برنامج فرعي ، وكما يلي :



شكل (1 . 13) أ ـ المخطط الانسيابي للبرنامج الفرعي الذي «يقرأ البيانات»



شكل (13.1) ب ـ المخطط الانسيابي للبرنامج الفرعي الذي ينجز حساب الراتب الكلي لكل مشتغل



شكل (13.1) جـ - المخطط الانسيابي للبرنامج الفرعي الذي يخرج بعض المعلومات المطلوبة عن كل مشتغل

مسألة (6) : أعد كتابة خوارزمية حل المسألة السابقة بشبه الشفرة مستخدماً أسلوب البرمجة الهيكلية.

الحـــــل :

خوارزمية الراتب الشهري الكلي

1 ـ ادخل البيانات.

2 ـ احسب الراتب الكلي.

3 ـ اخرج النتائج.

نهاية.

خوارزمية إدخال البيانات

1 ـ التمهيد لتصفير العداد.

2 ـ اقرأ راتب المشتغل الشهري.

3 ـ اقرأ عدد الساعات الاضافية لهذا المشتغل.

4 ـ اقرأ اجرة الساعة الاضافية لهذا المشتغل.

5 ـ أضف واحد الى عداد المشتغلين.

6 ـ هل العداد = 50؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (2).

7 ـ توقف.

عودة.

خوارزمية حساب الراتب الكلي

1 ـ التمهيد لتصفير العداد.

2 ـ احسب: المبلغ الاضافي الكلي = عدد الساعات الاضافية × اجر كل ساعة.

3 ـ احسب: الراتب الكلي = الراتب الشهري + المبلغ الاضافي الكلي .

4 ـ اضف واحد الى عداد المشتغلين.

5 ـ هل العداد = 50؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (2) .

6 ـ توقف .

عودة .

خوارزمية إخراج النتائج

1 ـ التمهيد لتصفير العداد .

2 ـ اكتب اسم المشتغل .

3 ـ اكتب راتبه الشهري (بالدينار) .

4 ـ اكتب المبلغ الاضافي الكلي (بالدينار) .

5 - اكتب راتبه الكلي لهذا الشهر (بالدينار) .

6 ـ أضف واحد الى عداد المشتغلين.

7 ـ هل العداد = 50؟ إذا كان الجواب بنعم فنفذ الخطوة التالية. وإذا كان الجواب بكلا فاذهب الى الخطوة رقم (2) .

8 ـ توقف .

عودة .

9. خلاصة :

قد تم تخصيص مفردات هذا الفصل لإرشاد القارىء المبتدىء على كيفية برمجة حاسبة الكترونية. فمفرداته قد اشتملت بالشرح بعض المبادىء الضرورية والمفاهيم الأساسية لمن يريد أن يستخدم الحاسبة الالكترونية كأداة في عمله عن طريق برمجتها. وفي اعتقادنا أن هذا الفصل سيساعد القارىء في تنمية وضبط جهده العقلي الذي يحتاجه عند برمجة حاسبة الكترونية.

يمكن إنجاز بعض المفاهيم والمبادىء الموضحة في هذا الفصل بالنقاط التالية:

أولاً : البرمجة هي عملية تحليل وتخطيط خطوات حل مسألة ومن ثم تشفيرها بلغة برمجة تستطيع الحاسبة الالكترونية من تنفيذ أوامرها وعباراتها. بعد ذلك يغذى البرنامج الى الحاسبة الالكترونية، وينفذ لاكتشاف الأخطاء وتصحيحها إن وجدت حتى يؤدي المهمة التي صمم من أجل حلها. لتصميم برنامج يجب التدرج في الإجابة على الأسئلة التالية:

(أ) هل تم ذكر خصائص المسألة بصورة واضحة وعلى نحو كامل؟

(ب) هل أنا أعرف طريقة لحل تلك المسألة؟

(جـ) هل أستطيع الحصول على طريقة حل لتلك المسألة من أشخاص آخرين أو بعض الكتب أو المجلات المتخصصة بالبرمجة؟

(د) كيف سأصمم خوارزمية حل تلك المسألة؟

ثانياً : هنالك ستة خطوات يجب اتباعها عند حل مسألة باستخدام الحاسبة الالكترونية.

(1) تعرّف المسألة بوضوح.

(2) هل المسألة هي مناسبة للحل بواسطة الحاسبة الالكترونية.

(3) تصميم خوارزمية الحل باستخدام طريقة المخطط الانسيابي أو شبه الشفرة. يمكن تحليل المسألة بأسلوب البرمجة الهيكلية.

(4) ترجم خطوات خوارزمية الحل الى احدى لغات الحاسبة الالكترونية.

(5) غذي البرنامج الى الحاسبة الالكترونية.

(6) ادرس نتائج البرنامج باختباره والتأكد من صحة عمله تحت جميع الشروط.

ثالثاً : الخوارزمية هي الانسياب المنطقي لمجموعة من الخطوات الدقيقة والتفصيلية لحل مسألة عن طريق الأخذ بنظر الاعتبار

كل الشروط والاحتمالات. يمكن كتابة الخوارزمية بالمخطط الانسيابي وبشبه الشفرة. ويمكن اتباع أسلوب البرمجة الهيكلية عند كتابة خوارزمية حل المسائل المعقدة. توفر الخوارزمية للمبرمج ستة فوائد راجعها في الفقرة رقم (4) .

رابعاً : المخطط الانسيابي هو طريقة وصف تصويرية لخطة تنفيذ خطوات حل مسألة معينة تمهيداً لبرمجتها بلغة حاسبة الكترونية مناسبة. يستخدم المخطط الانسيابي رموز ذات أشكال هندسية:

(أ) الرمز المتوازي الأضلاع لادخال البيانات أو إخراج النتائج .

(ب) الرمز المستطيل لمعالجة البيانات.

(جـ) الرمز المعين لاتخاذ القرار.

ترتبط تلك الأشكال الهندسية مع بعضها بخطوط ذات أسهم يدل اتجاهها على منطقة تنفيذ خطوات الحل.

خامساً : شبه الشفرة هي طريقة كتابة خوارزمية حل مسألة باستخدام كلمات اللغة التي يجيدها المبرمج (كالعربية أو الانكليزية أو الفرنسية. . .) حيث تجد تعليمات خوارزمية الحل عبارة عن جمل لغوية.

سادساً : البرمجة الهيكلية هو أسلوب تخطيط كتابة خوارزمية حل المسائل المعقدة جداً، حيث يتم تجزأتها الى مهمات

صغيرة، فتجد الخوارزمية تتكون من برنامج رئيسي وعدة برامج فرعية.

سابعاً : تتكون خوارزمية حل أي مسألة سواءً أكانت بسيطة أم معقدة من تشكيلة معينة لتركيبات أنماط منطقية ذات أنواع أربعة هي:

(أ) تركيبة التسلسل البسيط.

(ب) تركيبة الاختيار.

(جـ) تركيبة الانشوطة.

(د) تركيبة الانتقال.

ينصح أسلوب البرمجة الهيكلية بعدم استخدام تركيبة الانتقال لأنها تغير باستمرار مجرى تنفيذ البرنامج وبالنتيجة سيكون من الصعب تتبع الخطأ في البرنامج إن وجد.

ثامناً : صممت ستة أمثلة لتمرن القارىء على المفاهيم الأساسية ومبادىء البرمجة.

10. كلمات قد تعلمتها :

البرمجة	programming
الخوارزمية	Algorithm
المخطط الانسيابي	flow chart
شبه الشفرة	Pseudo Code
البرمجة الهيكلية	structured programming
رمز الإدخال/الاخراج	Input/Output symbol
رمز المعالجة	processing symbol
رمز القرار	decision symbol
تركيبة التسلسل البسيط	Simple sequence structure
تركيبة الاختيار	Selection structure
تركيبة انشوطة «اعمل أثناء»	loop structure (DO WHILE)
تركيبة انشوطة «اعمل حتى»	loop structure (DO UNTIL)
تركيبة الانتقال	Branch structure
برنامج رئيسي	main routine
برنامج فرعي	subrotine
عودة ؛ ارجع	Return
انشوطة	loop
عدّاد	counter
الرمز الطرفي	Terminal symbol
التمهيد (مثلاً تصفير العداد)	initialization

11. تمارين (1)

استخدم كلمات قد تعلمتها

(أ) البرمجة	(و) برنامج فرعي
(ب) الخوارزمية	(ز) عودة
(جـ) المخطط الانسيابي	(حـ) انشوطة
(د) شبه الشفرة	(ط) عدّاد
(هـ) البرمجة الهيكلية	(ى) رمز القرار

1 ـ ______ هي كلمة أساسية توضع في نهاية البرنامج الفرعي لتجعل الحاسبة الالكترونية ترجع الى البرنامج الرئيسي.

2 ـ يجب التمهيد للـ ______ بتصفيره. حيث أنه قد تم استخدامه للسيطرة على عدد ألمرات التي ستنفذ بها الانشوطة.

3 ـ يستخدم رمز معيني لاختبار تحقق شرط أو عدم تحققه عند رسم خوارزمية بالمخطط الانسيابي، ويدعى برمز الـ ______ .

4 ـ تعتبر ______ كالخريطة التي يستعين بها المبرمج عند كتابة برنامج حل مسألة.

5 ـ إن أسهل طريقة لكتابة خوارزمية حل مسألة معقدة هي بتجزئتها الى مهمات صغيرة تدعى كل منها بالـ ـــــــــــ .

6 ـ تدعى مجموعة عمليات تخطيط خطوات حل مسألة وتشفيرها بلغة برمجة وتنفيذها على حاسبة الكترونية ومن ثم تصحيح الأخطاء بالـ ـــــــــــ .

7 ـ عندما يستخدم المبرمج كلمات اللغة التي يجيدها فقط في كتابة الخوارزمية تدعى هذه الطريقة بـ ـــــــــــ . حيث تجد التعليمات عبارة عن جمل لغوية .

8 ـ في البرمجة يتم تكرار تنفيذ سلسلة من الخطوات بناء على اختبار شرط معين، يدعى ذلك بالـ ـــــــــــ .

9 ـ ـــــــــــ هو أسلوب يتبع عندما تكون المسألة معقدة جداً والذي ينصح بتجزئتها الى مهمات صغيرة.

10 ـ يفضل بعض المبرمجين استخدام طريقة وصف تصويرية في كتابة الخوارزمية حيث يتم استخدام رموز هندسية وخطوط ربط ذات اتجاه. تدعى هذه الطريقة بالـ ـــــــــــ .

صح أم خطأ ؟

1 ـ عشرون بالمئة فقط من وقت حل المسألة باستخدام الحسابة الالكترونية قد خصص لكتابة البرنامج .

2 ـ يجب أن تكون الخوارزمية عامة في صياغتها .

3 ـ تساعد الخوارزمية المبرمج على تجنب نسيان خطوات حل مسألة.

4 ـ تعتبر البرمجة الهيكلية مفيدة عندما تكون المسألة معقدة.

5 ـ الخوارزمية هي ترجمة حل المسألة الى لغة احدى الحاسبات الالكترونية.

6 ـ تنصح البرمجة الهيكلية باستخدام تركيبة الانتقال.

7 ـ في المخطط الانسيابي يستخدم الرمز المستطيل لادخال البيانات.

8 ـ شبه الشفرة هي احدى لغات برمجة الحاسبة الالكترونية.

9 ـ البرنامج الجيد هو الذي يعمل تحت جميع الشروط والظروف التي صمم ليواجهها.

10 ـ تكون للبرنامج الفرعي نقطة دخول واحدة ونقطة خروج واحدة.

أجب باختصار :

1 ـ لتتصرف على النحو السليم عند برمجة حل مسألة معينة، فإن هنالك أربعة أسئلة يجب التدرج في الإجابة عليهم الواحد تلو الآخر. اذكر تلك الأسئلة.

2 ـ قسم أساتذة برمجة الحاسبة الالكترونية وقت البرمجة. كيف تم التقسيم؟

3 ـ ما هي صفات البرنامج الجيد؟

4 ـ ماذا تستطيع الحاسبة الالكترونية إنجازه؟

5 ـ ما هي الأمور التي توفرها لنا الخوارزمية والتي جعلت المبرمجين متمسكين بها كأداة لتخطيط حل المسألة؟

6 ـ ما هي الأسباب التي تدعو الى استخدام المخطط الانسيابي عند كتابة الخوارزمية؟

7 ـ يتكون كل حل منطقي لمسألة باستخدام الحاسبة الالكترونية من تركيبات لأنماط منطقية. اذكر تلك التركيبات موضحاً اياها بالمخطط الانسيابي لكل منها.

8 ـ كيف تفكر عند برمجة حاسبة الكترونية؟

9 ـ ماذا تفضل عند كتابة خوارزمية الحل طريقة المخطط الانسيابي أم شبه الشفرة أو البرمجة الهيكلية؟ لماذا؟

10 ـ ما هي الأمور التي تشجعها طريقة البرمجة الهيكلية؟

اختار الجواب الصحيح :

1 ـ تستطيع الحاسبة الالكترونية من:
(أ) خزن واسترجاع ومقارنة البيانات.
(ب) إضافة وطرح وضرب وقسمة البيانات.
(جـ) التفكير.
(د) إنجاز كل من (أ) و(ب).

2 ـ البرمجة هي عملية
(أ) تخطيط خطوات حل مسألة على شكل خوارزمية.
(ب) تشفير خوارزمية الحل الى برنامج.

(جـ) تنفيذ البرنامج وتصحيح أخطائه إن وجدت.
(د) كل من (أ) و(ب) و(جـ).

3 ـ هل تعتبر احدى الخطوات التالية هي من خطوات حل المسألة باستخدام الحاسبة الالكترونية؟
(أ) تغذية البرنامج الى الحاسبة الالكترونية.
(ب) كتابة الخوارزمية.
(جـ) ترجمة الخوارزمية.
(د) كل الخطوات أعلاه.

4 ـ تعتبر أول خطوة عند حل مسألة باستخدام الحاسبة الالكترونية هي:
(أ) تصميم خوارزمية الحل.
() تعريف المسألة.
(جـ) ترجمة الخوارزمية.
(د) دراسة نتائج تنفيذ البرنامج.

5 ـ توجد أربعة من تركيبات الأنماط المنطقية هي:
(أ) التسلسل البسيط وانشوطة «اعمل حتى»، وانشوطة «اعمل أثناء»، والانتقال.
() التسلسل البسيط، والاختيار، والانشوطة، والانتقال.
(جـ) التسلسل البسيط، وانشوطة «اعمل حتى»، والاختيار والانتقال.
(د) التسلسل البسيط، وانشوطة «اعمل اثناء»، وانشوطة «اعمل حتى» والاختيار.

6 ـ قسّم أساتذة البرمجة الوقت المحدد لحل مسألة باستخدام الحاسبة الالكترونية. وقد خصصوا لكتابة الخوارزمية جزءً قدره ________ .

(أ) 80٪

(ب) 60٪

(جـ) 40٪

(د) 20٪

7 ـ أي من الطرق التالية تكتب بواسطتها الخوارزمية؟

(أ) البرنامج .

(ب) شبه الشفرة .

(جـ) المخطط الانسيابي .

(د) البرمجة الهيكلية .

8 ـ يكون رمز إدخال البيانات أو اخراج النتائج في المخطط الانسيابي عبارة عن شكل :

(أ) مستطيل .

(ب) متوازي أضلاع .

(جـ) دائري .

(د) معيني .

9 ـ عندما تكون المسألة التي يراد برمجتها معقدة فيفضل استخدام أسلوب ــــــــــ عند كتابة الخوارزمية .

(أ) الانشوطة .

(ب) شبه الشفرة .

(جـ) البرمجة الهيكلية .

(د) المخطط الانسيابي .

10 ـ يشترك كل من شبه الشفرة والمخطط الانسيابي على أن كلا منهما ــــــــــ .

(أ) هو اسم لعالم في البرمجة.

(ب) طريقة لكتابة الخوارزمية.

(جـ) لغة برمجة للحاسبة الالكترونية.

(د) أسلوب للعب مع الحاسبة الالكترونية.

الفصل الثاني

المبادىء الأساسية للبرمجة بلغة بيسك

1. المقدمــة
2. مفاهيم أساسية في لغة بيسك
3. أنواع البيانات
   اختبر نفسك
4. خزن البيانات ولهجات لغة بيسك
5. أنواع المتغيرات
   اختبر نفسك
6برمجة الرياضيات الحديثة بلغة بيسك
   اختبر نفسك
7. تركيب البرنامج بلغة بيسك
8. مسائل محلولة
9. خلاصــة
10. كلمات قد تعلمتها
11. تمارين (2)

1. المقدمة :

عرفت من الفصل السابق أسلوب التفكير الذي يتبعه المبرمجون لكتابة خوارزمية حل مسألة معينة. والتي تعتبر خطوة تمهيدية لبرمجة أي حاسبة الكترونية.

الآن سنكرس هذا الفصل لنعرفك على عناصر أو مركبات أو المبادىء الأساسية للغة بيسك. والسبب في ذلك، هو إعطائك معنى ونظرة أعمق عندما ستصف عبارات وأوامر لغة بيسك في الفصول اللاحقة من هذا الكتاب. وكذلك لنبين لك علاقة مكونات العبارة الواحدة مع بعضهم والعلاقة التفاعلية بين عبارات البرنامج المكتوب بلغة بيسك.

تكتب لغة بيسك بكلمات انكليزية بسيطة وواضحة المعنى. أما إجراء الحسابات فيتم بمعادلات مشابهة لما هو مستخدم في الجبر ولكن بأحرف انكليزية وأرقام عربية غبارية (0, 1, 2, 3, . . .) وعلامات العمليات الحسابية المعروفة. لذا كل ما نطلبه منك، عزيزي القارىء، هو معرفة الحروف الأبجدية الانكليزية وبعض الكلمات الانكليزية البسيطة لكي تبرمج بلغة بيسك. ستكون لدراستك هذا الفصل فوائد كثيرة تتجلى في إمكانيتك على :

1 ـ معرفة عناصر أي عبارة مكتوبة بلغة بيسك.

2 ـ التمييز بين نوعين من البيانات التي تعالجها برامج لغة بيسك.

3 ـ التمييـز بين أنواع المتغيـرات التي تسمى بها البيـانـات في بـرامـح لغـة بيسك.

4 ـ تـرجمة المعـادلات الريـاضية الى شكـل يمكن تنفيـذه في بـرامـج لغـة بيسك.

5 ـ معرفة أسبقية تنفيذ العمليات الرياضية المختلفة.

6 ـ توضيح تركيب أي برنامج بلغة بيسك.

7 ـ معرفة الفرق بين لهجات لغة بيسك من حيث خزنها للبيانات.

2. مفاهيم أساسية في لغة بيسك :

يمكن تعريف لغة برمجة الحاسبة الالكترونية، وبضمنها لغة بيسك، على أنها وسيلة اتصال بين الإنسان والحاسبة الالكترونية. يكون الاتصال على شكل برنامج (program) يتكون البرنامج بلغة بيسك من مجموعة رموز (characters) ومفردات (vocabulary) ترتبط فيما بينها وفق قواعد نحو (syntax) محددة لتكوين عبارات البرنامج الذي سيحل مسألة معينة.

إن الرموز التي تستخدم لتكوين عبارات لغة بيسك تسمى بالحروف الهجائية (Alphabet) . وهي تتكون من:

أولاً : **حروف اللغة الانكليزية** (A, B, C, ...) .

ثانياً : **الأرقام العشرية العربية** (0,1,2,3,4,5,6,7,8,g) .

ثالثاً : **رموز خاصة وعلامات العمليات الرياضية المعروفة**
(+، −، ★، ↑، ٪، /، ؟، ؛، . . .)

تتكون البرامج المكتوبة بلغة بيسك من مجموعة عبارات يجب فصل كل عبارة عن الأخرى في البرنامج الواحد بحيث تستخدم هذه العبارات لتخبر الحاسبة الالكترونية عن نوعية البيانات المدخلة والعمليات

الحسابية المنطقية لمعالجة البيانات وعن النتائج المطلوبة وكيفية إخراجها للمبرمج بشكل مفهوم. والعبارة (statement) هي الإيعاز في البرنامج. حيث تتألف العبارة، بصورة عامة، من العناصر التالية:

أولاً : رقم العبارة: توضع في بداية العبارة من جهة اليسار أرقام تميزها عن بقية عبارات البرنامج. بحيث تكون تلك الأرقام انعكاساً لتسلسل العبارة في البرنامج. ولتكون عملية الرجوع إلى هذه العبارة مباشرة عند تنفيذ البرنامج إذا لزم الأمر.

ثانياً : احدى الكلمات الأساسية (key word) المكونة للغة بيسك، والتي تخبر الحاسبة الالكترونية ما الذي ستنجزه في هذه العبارة. في بعض العبارات، تستخدم كلمتان أساسيتان في العبارة الواحدة.

ثالثاً : معلمة (parameter) : وهي إما متغير أو ثابت أو تعبير رياضي. وتعتبر المعلمة وسطاً لتنفيذ العملية المقترحة من قبل الكلمة الأساسية في العبارة.

مثال: فيما يلي برنامج بسيط مكتوب بلغة بيسك وقد أشرنا عليه بعض المفاهيم الأساسية المذكورة أعلاه:

كلمة أساسية ↓

```
عبارة بلغة بيسك → 10  READ   A1, A2
                  20  LET    A3 = A1 + A2     ← معلمة
                  30  PRINT  A3
                  40  DATA   -310, 99
                  50  STOP
                  60  END
                  ↑
              رقم العبارة
```

3. أنواع البيانات :

إن هـدف لغة بيسك هـو معـالجـة بيـانـات رقميـة (numeric data) وبيـانات وصفيـة (textual data) . والبيـانـات هي المعلومـات الخـام التي تزود بها الحاسبة الالكتـرونية لتختـزنها أو تعـالجها ريـاضياً للحصـول على النتـائـج المـطلوبـة. تعـرف الأعـداد على أنهـا ثـوابت رقميـة (numeric constants) أو بيـانـات رقميـة، وتعـرف أي سلسلة من الـرمـوز على أنهـا ثـوابت مقطعيـة (string constants) أو بيانـات وصفية. سنتنـاول بـالشـرح نوعي البيانات المذكورة أعلاه:

أولاً : البيانات الرقمية :

تـزود الحاسبـة الالكترونيـة باستخـدام لغة بيسـك الأعداد على أنهـا بيـانـات رقميـة. تكـون للبيـانـات الـرقميـة عـدة أشكـال منهـا الصحيحة والحقيقية.

مثـــال :

715	− 61	+ 5893	(أعداد صحيحة)
− 41.07	+ 21.34	− 95.82	(أعداد حقيقية)

لاحـظ أن النقطة المـوجودة بين أرقـام العدد الصحيـح هي الفـارزة

العشرية التي عهدتها. وإذا حذفت العلامة (+) فالعدد موجب. ولا يسمح بوجود فارزة ضمن العدد.

تطبع النتائج في لغة بيسك أيضاً باستخدام أعداد صحيحة أو حقيقية. بحيث تجد الأعداد السالبة تسبق بعلامة سالبة والأعداد الموجبة تسبق بفراغ. أما إذا كان العدد كبيراً جداً أو صغيراً جداً فإن الحاسبة الالكترونية تخرج النتائج على شكل اسي حيث تجد الرمز (E) يوضع للدلالة على الرفع للأساس (10) ؛

مثال :

شكل اعتيادي	شكل اسي
0.00083	8.3 E−4
463000	4.63 E+5
0.0011	−1.1 E−3

يمكن إدخال البيانات باستخدام الشكل الاسي.

تعرّف لغة بيسك، ما اتفق عليه، أن أكبر ثابت رقمي هو (1 E+38) وأصغر ثابت رقمي هو (1 E−38) . واضعين في الحسبان أن تلك القيم تعتمد على آلة الحاسبة الالكترونية نفسها وليس على لغة بيسك، ولكن تمثل الأرقام المذكورة أعلاه المدى المتعارف عليه من البيانات الرقمية التي صممت لغة بيسك لمعالجتها. هذا يعني لا توجد حاسبة الكترونية تستخدم لغة بيسك ولا تستطيع معالجة ثابت رقمي مقداره:

± 100.000,000,000,000,000,000,000,000,000,000,000,000,000

أما دقة (precision) البيانات الرقمية فيجب أن لا تقل عن ستة

أرقام عشرية. هذا يعني أنه سيخزن العدد (123456789) بشكل تقريبي على أنه (000 456 123) .

ثانياً : البيانات الوصفية :

هي تشكيلة من رموز متجاورة (string) . تجهز البيانات الوصفية إلى الحاسبة الالكترونية باستخدام لغة بيسك بوضعها داخل قوس الاقتباس (" "). يمكن أن يظهر أي رمز يمكن طبعه بلوحة مفاتيح الحاسبة الالكترونية ضمن البيانات الوصفية ما عدا علامة الاقتباس لأنها تغلق مقطع الرموز تلقائياً عند ظهورها حتى داخل علامة الاقتباس. يعتبر الفراغ أيضاً من ضمن الرموز المسموح باستخدامها في البيانات الوصفية.

مثـــال :

"My Name Is"

"   "

"AUGUST 14th. 1986"

في الفرع الأخير من المثال أعلاه قد استخدمت ثوابت رقمية ضمن البيانات الوصفية، وبكن لاحظ عدم إمكانية استخدام العمليات الرياضية على البيانات الوصفية.

هنالك استثناءات في لغة بيسك تسمح بتعريف البيانات الوصفية بدون استخدام علامتي الاقتباس، كما في عبارة بيانات (DATA) وكاستجابة لتنفيذ عبارة اطبع (PRINT) التي تتطلب إدخال بيانات وصفية.

اختبر نفسك :

1 ـ اذكر العلاقة بين البرنامج الرموز المفردات وقواعد النحو.

2 ـ مما تتكون العبارة في لغة بيسك؟

3 ـ اذكر أنواع البيانات التي صممت لغة بيسك لمعالجتها بحاسبة الكترونية؟

4 ـ هنالك ثلاثة أشكال يمكن أن يأخذها الثابت الرقمي عند إدخاله أو إخراجه من حاسبة الكترونية باستخدام لغة بيسك، اذكر تلك الأشكال مع الأمثلة؟

5 ـ ما هي البيانات الوصفية؟ وكيف تعرف في عبارات برنامج بلغة بيسك؟

4. خزن البيانات ولهجات لغة بيسك :

تعتبر الذاكرة الرئيسية الأساسية في الحاسبة الالكترونية. حيث تتكون الذاكرة الرئيسية من آلاف الخلايا تدعى كل خلية بالبايت (byte) بحيث يحتفظ البايت بكمية محدودة من المعلومات.

تمثل البيانات الرقمية ضمن عدد محدود من البايت، لذلك تكون هنالك حدود لمدى (range) الرقم الذي يمكن خزنه. وكذلك حدود لدقة (prescision) عدد الأرقام المؤثرة (significant figures) التي يمكن خزنها.

تخزن البيانات الوصفية في الحاسبة الالكترونية بتخصيص بايت واحد لكل رمز. ويكون عدد البايت محدود في الذاكرة الرئيسية، فهنالك حدود لأكبر عدد من الرموز المسموح تواجدها في المقطع (string) الواحد.

لا يستطيع المبرمج تغيير حدود خزن البيانات الرقمية أو دقتها أو عدد الرموز في البيانات الوصفية لأن ذلك متأصلاً في لهجات (dialects) لغة بيسك المختلفة. والمقصود بلهجات لغة بيسك هو الاجتهاد الموجود بين مصنعي الحاسبات الالكترونية لتصميم شكل للغة بيسك يناسب تطبيقات معينة.

لاحظ الجدول التالي الذي يقارن الحدود المفروضة على خزن البيانات من بين أربع لهجات للغة بيسك.

نوع الحاسبة الالكترونية(اللهجة)	أصغر رقم موجب	أكبر رقم موجب	الدقة (بالأرقام المؤثرة)	أكبر عدد من الرموز في المقطع الواحد
CBM PET	2.93×10^{-39}	$1.7\times10^{+38}$	(9) أرقام مؤثرة	255
MSI	1.0×10^{-99}	$9.99\times10^{+99}$	(9) أرقام مؤثرة	128
PRIME	1.0×10^{-99}	$9.99\times10^{+99}$	(13) رقم مؤثر	160
APPLESOFT 11	1.0×10^{-37}	$9.99\times10^{+37}$	(9) أرقام مؤثرة	255

مثال : لنأخذ بنظر الاعتبار خزن البيانات التالية باستخدام حاسبة الكترونية شخصية تستخدم لغة بيسك بلهجة (APPLESOFT 11) .

(أ) $(0.0835 \times 10^{-33}) = 8.35 \times 10^{-35}$

من الجدول أعلاه، أن أصغر رقم يمكن تمثيله بهذه اللهجة هو (1.0×10^{-37}) لذلك يمكن أن يخزن الرقم اعلاه بشكل (8.35×10^{-35}) لأنه لم يتجاوز الحد الأدنى.

(ب) $(0\ 0835 \times 10^{-35}) = 8.35 \times 10^{-38}$

إن العدد أعلاه قد تجاوز أدنى حد للثوابت الرقمية المسموح بها باستخدام اللهجة أعلاه، لذلك سيحدث ما يدعى «دون الحد الأدنى» (UNDER FLOW) يمكن أن يخزن العدد أعلاه بقيمة تساوي الحد الأدنى المسموح به.

(جـ) $(98765.4321\times10^{+33}) = 9.87654321\times10^{+37}$

إن العـدد أعلاه لـم يتجـاوز أكبر حـد للبيانـات الرقميـة المسموح بهـا في هـذه اللهجة وهـو $(9.99\times10^{+37})$، لذلـك فـإنـه سيخـزن بشكـل $(9.87654321\times10^{+37})$.

(د) $(98765.4321\times10^{+34}) = 9.87654321\times10^{+38}$

إن العدد أعلاه قـد تجاوز الحـد الأعلى للبيانـات الرقميـة التي يمكن أن تخـزن بهـذه اللهجـة من لغـة بيسـك والـذي هـو $(9.99\times10^{+37})$ لذلك فإنه لا يمكن تمثيله بـاستخدام هـذه اللهجة لحـدوث ما يـدعى «فـوق الحد الأدنى» (OVERFLOW). يمكن أن يخـزن العـدد أعـلاه على أنه الحد الأعلى المسموح به $(9.99\times10^{+37})$.

(هـ) (0.0987654321 987)

إن دقة هذه اللهجة هي لتسعة أرقـام مؤثرة، لـذا سيخزن العـدد أعلاه بتسعة مراتب عشرية فقط بعد العلامة العشرية (0,098765432)

(و) "AUGUST 16th. 1986"

يمكن خزن الثابت الـوصفي أعلاه لأنـه لا يتجاوز الحـد الأعلى لعدد الـرمـوز المسـوح بهـا في المقـطع الـواحـد والتي هي (255) لهـذه اللهجـة. وفي حالـة تجاوز الحـد الأعلى فإن الحـاسبـة الالكتـرونيـة ستظهر رسالة تبين للمستفيـد أن عدد الـرموز قـد تجاوز الحـد الأعلى وأن هذا الثابت الوصفي سوف لن يخزن فيها.

من الأمثلة أعـلاه يجب أن نـلاحظ أن لأي لهجـة في لغـة بيســك، وبغض النظر عن نوع الحاسبة الالكتـرونية المستخـدمة، هنـالك عـدة أمور عامة:

أولاً : يمكن خزن مدى محدود من البيانات الرقمية.

ثانياً : يمكن تمثيل البيانات الرقمية بدقة محدودة.

ثالثاً : تمثل البيانات الوصفية بطول مقطع محدد مسبقاً.

5. أنواع المتغيرات :

يتم التعامل بالرموز في علم الجبر وفق قواعد موضوعة مسبقاً ولأن الرمز يمكن أن يأخذ طيف واسع من القيم الرقمية فإن الطرق الجبرية تكون عامة بطبيعتها.

اقتبس علماء الحاسبات الالكترونية الفكرة أعلاه عند تصميم لغة برمجة كلغة بيسك. فللوصول الى بيانات مخزونة في الذاكرة، يجب تسمية مناطق الذاكرة التي تحتوي تلك البيانات برمز، والذي يدعى المتغير (variable). وقد سميت متغير لإمكانية استبدال القيمة الرقمية التي يمثلها باستمرار نتيجة لتنفيذ عبارات برنامج بيسك.

تستطيع لغة بيسك من التعامل مع ثلاثة أنواع من المتغيرات، منها البسيط والنوع الثالث يكون ذي أبعاد سنوضح كل نوع على حدة:

أولاً : المتغيرات الرقمية (numeric variables)

يعرّف المتغير الرقمي في لغة بيسك إما برمز واحد من حروف الهجاء الانكليزية (26) أو برمزين؛ يأخذ الأول من حروف الهجاء الانكليزية والثاني من الأرقام العشرية (10). وبذلك يوجد (286) متغير رقمي في لغة بيسك. تستخدم المتغيرات الرقمية للإشارة الى مناطق

الـذاكرة التي تخـزن البيـانـات الـرقميـة. من المستحين أن يمهـد للمتغيـر الرقمي بمساواته للصفر، لتهيئـة أماكن الـذاكرة لاستقبـال بيانـات البرنـامج الجديد.

A	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
B	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
C	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
D	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
E	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9
F	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
G	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9
H	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9
I	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9
J	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9
K	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9
L	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9
M	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
N	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9
O	O0	O1	O2	O3	O4	O5	O6	O7	O8	O9
P	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
Q	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
R	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9
S	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
T	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
U	U0	U1	U2	U3	U4	U5	U6	U7	U8	U9
V	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9
W	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9
X	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9
Z	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9

ثانياً : المتغيرات الوصفية (textual variables)

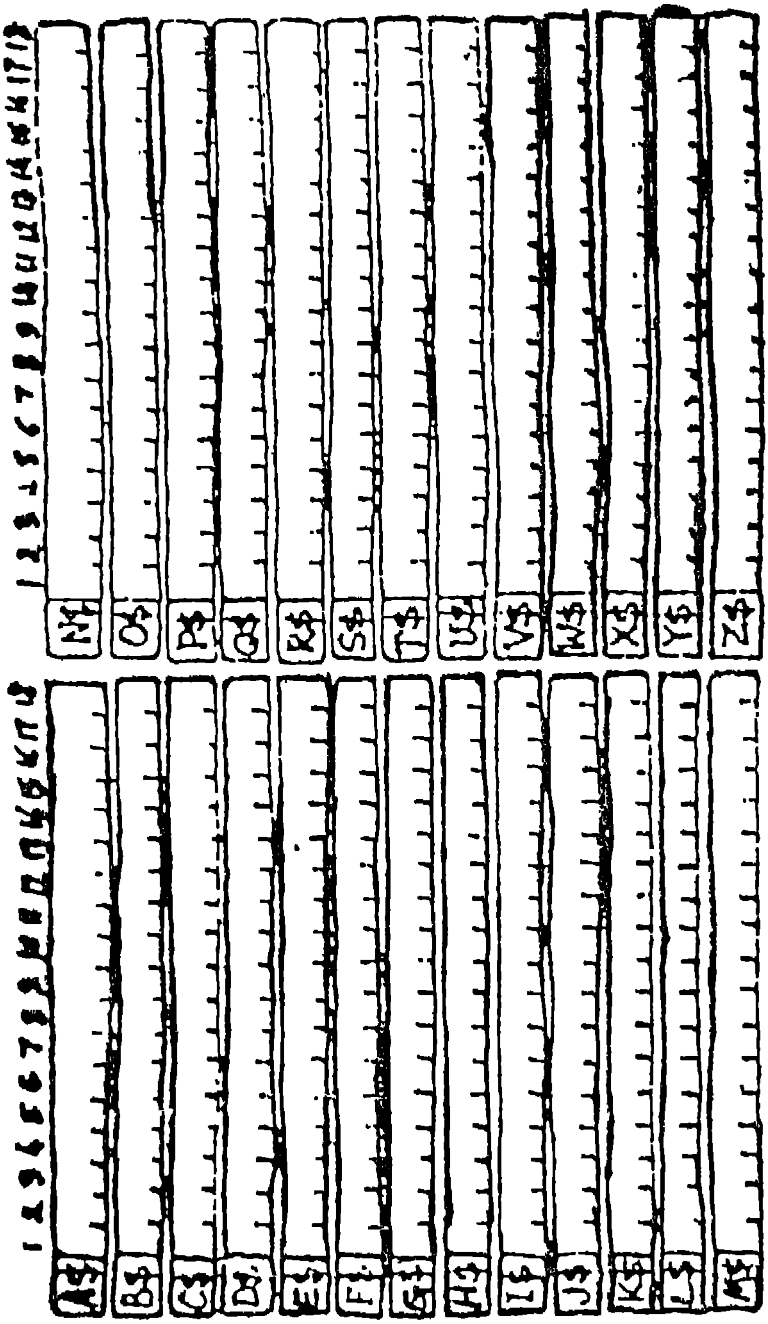
تستخدم المتغيرات الوصفية لاستدعاء البيانات الوصفية المخزونة في مواقع الذاكرة لكي تستخدم في برنامج بلغة بيسك. يعرّف المتغير الوصفي برمزين يكون الأول هو أحد حروف الهجاء الانكليزية متبوعاً برمز علامة الدولار ($) وبذلك فإن لدينا (26) اسماً لمتغير وصفي يمكن بواسطتها تمييز البيانات الوصفية في برنامج بلغة بيسك.

كل لهجة في لغة بيسك، تسمح بعدد محدد من الرموز التي يمثلها المتغير الوصفي. فكما لاحظنا من الفقرة (4) من هذا الفصل، تسمح بعض اللهجات بخزن (255) رمزاً مسموح به هو (18) رمزاً. أخيراً، من الأفضل التمهيد للمتغير الوصفي لكي تهيء مواقع الذاكرة لاستقبال رموز البيانات الوصفية لبرنامج بيسك جديد.

ثالثاً : المتغيرات ذات الأبعاد (Array variables)

في تطبيقات كثيرة، نحتاج برامج بلغة بيسك لمعالجة قائمة من البيانات سواء الرقمية أو الوصفية. لذا فمن المستحين، أن تكون من قابليات لغة بيسك استخدام اسم متغير واحد للإشارة الى مجموعة مترابطة من البيانات عوضاً عن استخدام متغير منفصل لكل حد من تلك البيانات المترابطة.

إن المتغير ذو الأبعاد هو مجموعة متغيرات لها نفس الاسم ولكن تتميز باستخدام معلمة (parameter) بين قوسين. يمكن أن يمثل المتغير ذو الأبعاد كلاً من البيانات الرقمية أو البيانات الوصفية. يعرّف المتغير ذو الأبعاد الذي يمثل البيانات الرقمية بأحد حروف الهجاء الانكليزية متبوع بمعلمة موضوعة داخل قوسين. أما الذي يمثل البيانات الوصفية فيتكون



من أحد رموز الحروف الهجائية الانكليزية مع علامة الدولار ($) متبوعاً بمعلمة موضوعة داخل قوسين.

مثـــال :

H(6)	E(N)	Z(10)	متغيرات ذات أبعاد تمثل بيانات رقمية
K$ (100)	C$ (K−1)	Y$(9)	متغيرات ذات أبعاد تمثل بيانات وصفية

لا يمكن استخدام نفس المتغير لتسمية متغير بسيط ومتغير ذو أبعاد في نفس البرنامج المكتوب بلغة بيسك.

يجب إعلان حجم المتغير ذو الأبعاد بعبارة الأبعاد (DIM) التي سيأتي ذكرها في فصل لاحق من هذا الكتاب.

تستطيع لغة بيسك معالجة متغيرات لها بعدين. حيث تجد المتغير ذو البعدين يتكون من اسم المتغير متبوع بقوسين فيهما معلمتين تفصلهما فارزة. كما في المثال التالي:

مثـــال :

G(91,13)	B(N,N+M/2)	Z(10,7)	متغيرات ذات بعدين تمثل بيانات رقمية
S$(10,70)	A$(K−1, K*N)	X$(16,11)	متغيرات ذات بعدين تمثل بيانات وصفية

تستخدم المتغيرات ذات البعدين في برنامج بلغة بيسك لمعالجة بيانات على شكل جدول أو ما يدعى بالمصفوفة (Matrix) ولذلك فائدة لا تخفى على المشتغلين بالرياضيات العليا.

اختبر نفسك :

1 ـ ما المقصود بمدى البيانات الرقمية ودقة البيانات الرقمية؟ اذكر المدى والدقة المتعارف عليها.

2 ـ هل يكون طول المقطع في البيانات الوصفية ثابتة لكل الحاسبات الالكترونية التي تستخدم لغة بيسك؟ اسند إجابتك بأمثلة.

3 ـ كم نوع من المتغيرات تستطيع لغة بيسك أن تتعامل معها؟ اذكرها باختصار.

4 ـ كيف يتم تمييز المتغير الوصفي عن المتغير الرقمي في لغة بيسك؟

5 ـ ما حاجتنا للمتغير ذو الأبعاد؟ كم نوع تستطيع لغة بيسك من معالجته؟ اذكر أمثلة لكل نوع.

6. برمجة الرياضيات الحديثة بلغة بيسك :

تمكّن لغة بيسك المبرمج من استخدام الحاسبة الالكترونية لمعالجة البيانات الداخلة إليها بعمليات رياضية وإجراءات متنوعة منها :

أولاً : العمليات الحسابية : مثل الجمع والطرح والضرب والقسمة والرفع لأس.

ثانياً : العمليات المنطقية : مثل عملية (AND) وعملية (OR) .

ثالثاً : عمليات المقارنة : مثل أكبر من (>) أو أصغر من (<) أو لا يساوي (<>) .

رابعاً : عمليات مزج البيانات الوصفية .

خامساً : تحريك البيانات الرقمية أو الوصفية من مواقع خزن أولية الى مواقع خزن أخرى بعد إجراء العمليات المطلوبة عليها .

سادساً : توقيت وتوجيه إجراء العمليات الرياضية على البيانات .

لإجراء العمليات أعلاه فإننا نحتاج الى تعابير رياضية (mathematical expressions) . والتعبير الرياضي هو وسيلة لحساب قيمة لمتغير

عن طريق ربط المتغيرات والثوابت بعمليات رياضية معينة. تستطيع لغة بيسك التعامل مع عدة أنواع من التعابير الرياضية، أهمها:

أو : التعابير الجبرية Algebraic expression

وتدعى أيضاً التعابير الرقمية أو التعابير الحسابية. تبنى التعابير الجبرية من العمليات الحسابية التالية:

الرفع لأس	↑ أو ** أو ^
القسمة	/
الضرب	*
الجمع	+
الطرح	−

الرمز المذكور ازاء كل عملية حسابية يمثل ما متعارف عليه عند البرمجة بلغة بيسك، وكما في المثال التالي:

مثـال :

تعابير جبرية	تعابير بلغة بيسك
x^3	X ↑ 3
$x \div y$	X/Y
$x . y$	X * Y
$x \div w + z . y$	X/W + Z * Y

يمكن إدخال الأقواس على التعابير الجبرية لتوضيح المطلوب إنجازه بتلك التعابير، وكما في الأمثلة التالية:

مثـــال :

تعبير جبري	تعبير بلغة بيسك
$\frac{x+y}{w \cdot z}$ $\sqrt{x^2 - 4 \cdot y \cdot z}$	(X + Y)/(W * Z) ((X ↑ 2) − (4 * Y * Z)) ↑ 0.5

في لغـة بيسـك، إذا احتـوى التعبيـر الجبـري على أكثـر من عمليـة حسـابية فسيتم تنفيـذ العمليـات الحسـابيـة حسب نـظام أسبقيـة، كمـا في التسلسل التالي :

الرفع لأس	أعلى أسبقية
الضرب أو القسمة	
الجمع أو الطرح	أدنى أسبقية

حيث أعـطيت لعملية الـرفع لأس أعلى أسبقيـة تليـه كـلًا من عمليـة الضـرب أو القسمـة ومن ثم يـأتي أخيـراً عمليتي الجمـع أو الـطرح. إذا ظهـرت في التعبيـر الجبـري العمليـات ذات الأسبقيـة نفسهـا (كـالجمـع والـطرح) فيتم تنفيذ العمليـة التي تظهـر أولًا من جهـة اليسـار الى اليمين. إذا احتـوى التعبير الجبـري على أقـواس فـإن العمليـات المـوجـودة داخـل القوس تنفذ أولًا وحسب نـظام الأسبقية أعـلاه، ومن ثم تنفذ بقيـة عمليات التعبير الجبرية الموجودة خارج القوس.

مثـال :

خزن النتيجة	العملية	الأسبقية	تعبير جبري بلغة بيسك
R1	X ↑ 2	1	((X ↑ 2) – (4⋆Y⋆Z)) ↑ 0.5
R2	4⋆Y⋆Z	2	
R3	R1 – R2	3	
R4	R3 ↑ 0.5	4	

ثانياً: التعابير المقطعية string expressions

- إن فكـرة التعبير المقـطعي محـدودة في لغـة بيسـك، حيث نجـدهـا بـأبسط صورهـا من مساوات متغيـر وصفي بمتغير وصفي آخـر أو مسـاوات متغير وصفي بثابت مقطعي، وكما في المثال التالي:

مثـال :

Y $ (9) = S $

C $ = "AUGUST 22th. 1986"

في بعض لهجـات لغـة بيســك، يمكن جعـل ثـابتين مقـطعيـن يتسلسلان في الظهور، وتدعى هـذه العملية بـالوصـل (concatenation) . ويرمز لها بعلامة الزائد (+) .

مثـال :

X $ = Y $ (9) + C $

ثالثاً : تعابير متراجحة Relational expressions

يتـألف تعبير المتـراجحـة من تعبيـرين جبـريين أو تعبيـرين مقـطعيين تفصل بينهما عـلامة متـراجحة. تتقبـل لغة بيسـك ستة عمليـات متراجحـة هي:

المعنى	العملية المتراجحة
المساوات	=
لا يساوي	< >
أصغر من	<
أصغر من أو يساوي	< =
أكبر من	>
أكبر من أو يساوي	> =

تستخدم تعابير المتراجحة في العبارة الشرطية "IF... THEN" ، التي سيأتي شرحها في فصل لاحق. يمكن مزج اكثر من تعبير متراجح في عبارة شرطية واحدة باستخدام احدى العمليتين المنطقيتين (OR) أو (AND) .

اختبر نفسك :

1 ـ ما هو التعبير الرياضي؟ وما هي أنواعه؟

2 ـ ما هي العمليات الرياضية التي للغة بيسك القابلية على إنجازها؟

3 ـ هل تختلف التعابير الرياضية في الجبر عما هو مستخدم في لغة بيسك؟ كيف؟

4 ـ كيف يتم تنفيذ العمليات الحسابية في تعبير جبري؟

5 ـ أعد صياغة المعادلة الجبرية التالية بشكل تتقبله لغة بيسك. ومن ثم نفذ عملياته الحسابية حسب نظام الأسبقية المعروف في لغة بيسك.

$$x = 500\ \sqrt{a^3 + b^7} \div c\ .\ d$$

7. تركيب البرنامج بلغة بيسك :

يتألف برنـامج البيسـك من أسطر مـرتبة بـرقم العبارة، بحيث يكـون آخـر سطر هـو لعبارة نهـاية البـرنـامـج (END) يتم تنفيـذ أسـطر البـرنـامـج بالتسلسل وابتداء من أول سطر حتى يتحقق أحد الأمور التالية :

- طلب تنفيذ عبارات أخرى بواسطة عبارة سيطرة.
- حدوث شرط استثنائي بحيث يؤدي الى توقف تنفيذ البرنامج.
- تنفيذ عبارة التوقف (STOP) أو عبارة نهاية البرنامج (END) .

يمكن استخدام الفراغات في برنامج بيسك لتحسين مظهره وإمكانيـة قـراءته. لا تؤثـر الفراغـات على تنفيذ بـرنـامـج البيسـك إذا لم تـظهـر في الأماكن التالية :

- في بداية العبارة.
- ضمن حروف الكلمة الأساسية للعبارة.
- ضمن رموز الثابت الرقمي.
- ضمن رقم العبارة.
- ضمن اسم الدالة (Function) أو اسم المتغير.
- ضمن عـلامـة المتـراجحـة ذات الـرمـزين، مثــل (<=) أو (>=) أو (<>).

يجب أن تسبق كـل الكلمات الأسـاسية في بـرنامـج البيسـك بفـراغ واحد أو أكثر.

يبـدأ كل سـطر برقم عبـارة. والذي يكـون عدد صحيـح موجب. لا تأثير للأصفار التي يبتـدأ بها رقم العبـارة. تكون أرقـام العبارات تصـاعديـة بحيث يعـطى أعلى رقم لعبارة نهـاية البـرنامـج (END) . يمكن أن يحتوي السطر الواحد على (72) رمزاً كحد أعلى.

مثال: دقق في برنامج البيسك التالي تـركيبة البـرنامـج المذكـورة أعلاه من حيث تكون البرنامج من أسطر مرقمة وأمكنة الفـراغات في البـرنامـج ومكونات العبارة الواحدة.

```
10  READ A1, A2, A$
20  LET A3 = (A1 + A2) ↑ 0.5
30  PRINT A$ , A3
40  DATA – 310, 99, "THE RESULT IS"
50  STOP
60  END
```

8. مسائل محلولة :

مسألة (1) : تعرّف على نوع كل حد من البيانات التالية، وسمي كل منها بمتغير مناسب:

(أ) "IRAQ"

(ب) 910-

(جـ) 72 E−3

(د) "GOOD PROGRAM"

(هـ) 531+

(و) 40.71−

(ز) "BASIC"

(حـ) 862 E+4−

(ط) 15.15+

(ى) AUGUST 24th. 1986"

الحـــل :

ت	البيانات	نوعها	اسم متغير
أ	"IRAQ"	ثابت مقطعي	I$
ب	−910	ثابت رقمي ـ عدد صحيح	N
جـ	72 E−3	ثابت رقمي ـ عدد اسي	X
د	"GOOD PROGRAM"	ثابت مقطعي	G $
هـ	+531	ثابت رقمي ـ عدد صحيح	A
و	−40.71	ثابت رقمي ـ عدد حقيقي	F

ز	"BASIC"	ثابت مقطعي	B$
ح	862 E+4	ثابت رقمي ـ عدد اسي	C
ط	−15.15	ثابت رقمي ـ عدد حقيقي	Y
ي	"AUGUST 24th. 1986"	ثابت مقطعي	U $

مسألة (2) : تستطيع حاسبة الكترونية خزن أعداد صحيحة ضمن المدى (43854+) إلى (43854−) وخزن أعداد حقيقية ضمن المدى ($\pm 01\times 10^{-40}$) إلى ($\pm 0.99\times 10^{+40}$) بدقة قدرها (6) أرقام مؤثرة. وتسمح بخزن ثابت مقطعي له (18) رمزاً فقط. كيف ستخزن البيانات التالية:

(أ) $-8765.91 \times 10^{+35}$

(ب) "I LIKE PROGRAMMING IN BASIC"

(جـ) -9.01 E $-$ 8

(د) + 19992

(هـ) $+ 6529.4 \times 10^{+38}$

(و) "TYPE YOUR NAME IN C.L."

(ز) 384.6 E + 19

(حـ) − 55551

(ط) − 0.888888889

(ى) $73529.84 \times 10^{+30}$

الحــــل :

(أ) $- 0.876591 \times 10^{+39}$

(ب) لا يخزن الثابت المقطعي لأنه يحتوي على (26) رمزاً.

(جـ) $-0.901\ E-9$

(د) $+0.19992\times 10^{5}$

(هـ) سيحدث دون الحد الأدنى (overflow) لأن الثابت الرقمي ($0.65929\times 10^{+42}$) قد تجاوز الحد الأعلى المسموح به للخزن.

(و) لا يخزن الثابت المقطعي لأنه يحتوي على (23) رمزاً والمسموح به هو (18) رمزاً فقط.

(ز) $0.3846\ E+22$

(حـ) $-0.5551\times 10^{+5}$

(ط) -0.888888 ، قد حذفت الأرقام الثلاثة الأخيرة لأن دقة هذه الحاسبة هي (6) أرقام مؤثرة فقط.

(ى) $0.735298\times 10^{+35}$

مسألة (3) : تعرّف على أسماء المتغيرات المسموح بها في لغة بيسك مما يلي. واذكر سبب كون بعضها غير مسموح به.

(أ) 4Z4

(ب) XY

(جـ) 100 D

(د) 2R$

(هـ) S2$

(و) H$

(ز) F3 (10, 30)

(حـ) B$(K. N−1)

(ط) AB̧ (19)

(ى) C$(3)

الحـــل :

(أ) غير مسموح به، لوجود رقم عشري قبل حرف الهجاء الانكليزي (Z).

(ب) غير مسموح بـه، لوجـود رمزين من حـروف الهجاء الانكليـزي لنفس المتغير.

(جـ) غيـر مسموح بـه، لوجـود عدد عشـري قبل حـرف الهجاء الانكليـزي (D) .

(د) غير مسموح به، لوجود رقم عشري قبل اسم المتغير الوصفي.

(هـ) غير مسموح به، لوجود رقم عشري بين حـرف الهجاء الانكليـزي (S) وعلامة الدولار.

(و) مسموح به.

(ز) غيـر مسموح بـه، لوجـود رقم عشري بعـد حرف الهجـاء الانكليـزي (F) .

(حـ) مسموح به.

(ط) غيـر مسموح بـه، لوجـود رمزين من حـروف الهجاء الانكليـزي بدلاً من حرف واحد.

(ى) مسموح به.

مسـألة (4) : تعـرّف على الثوابت المسمـوح استخدامهـا في لغـة بيسـك. واذكر السبب إذا لم تكن من المسموح بها.

(أ) 89,76	(و) 721 +
(ب) "MY NAME IS"	(ز) 1.01 −
(جـ) 60%	(حـ) 3.56 E − 5
(د) 3.412 E − 0.5	(ط) "4197, SHAAB CITY"
(هـ) I$	(ى) 123.78 − +

الحـــل :

(أ) غير مسموح به، لوجود فارزة بين أرقام العدد.

(ب) غير مسموح به، لفقدان أحد قوسي الاقتباس.

(جـ) غير مسموح به، لوجود علامة النسبة المئوية (٪).

(د) غير مسموح به، لكون الأس عدد حقيقي وليس عدداً صحيحاً.

(هـ) غير مسموح به، لأنه اسم متغير وصفي وليس ثابتاً مقطعياً.

(و) مسموح به.

(ز) مسموح به.

(حـ) مسموح به.

(ط) مسموح به، إذا كانت حاسبتك تستخدم لهجة تسمح باستخدام الفارزة ضمن الثابت المقطعي.

(ى) غير مسموح به، لوجود علامتين حسابيتين لنفس الثابت الرقمي.

مسألة (5) : كيف تكتب المعادلات الرياضية التالية في لغة بيسك؟

(أ) $C = \sqrt{a^2 + b^2}$

(ب) $t = \left[\frac{(a-b)^2}{(c+a)^m} - \frac{\sqrt{b+c}}{7\,(p+q)^K} \right]^{1/n}$

(جـ) $v \geqslant x + y \div z$

(د) $x \leqslant (u+v)^{n-1}$

(هـ) $K = (8n)^{1/3}$

(و) $d = \sqrt{b^2 - 4 . A . C}$

(ز) $w = x^b \div (y - z)$

(حـ)

$$z = \frac{(x_1 + x_2)^m (y_1 + y_2)^n}{(x_1/y_1)^{m+n} (x_2/y_2)^{m-n}}$$

(ط)

$$y = \frac{a_1 - a_2 x + a_3 x^2 - a_4 x^3 + a_5 x^4}{c_1 - c_2 x + c_3 x^2 - c_4 x^3}$$

(ى)

$$s = \frac{2(p/q)^{K-1}}{(r-3t)^{1/m}}$$

الحـــل :

(أ) C = (A ↑ 2 + B ↑ 2) ↑ 0.5

(ب)

(((A−B) ↑ 2/ (C+A) ↑ M) − ((B+C) ↑ 0.5/7* (P+Q) ↑ K)) ↑ (1/N)

(جـ) V > = X + Y/Z

(د) X < = (U + V) ↑ (N−1)

(هـ) K = (8 * H) ↑ (1/3)

(و) D = (B ↑ 2 − 4 * A * C) ↑ 0.5

(ز) W = X ↑ B/(Y − Z)

(حـ)

Z =(((X1+X2) ↑ M* (Y1+1/2) ↑ N)/ ((X1/Y1) ↑ (M+N)

* (X2/X2) ↑ (M−N)) (1/MN)

(ط)

Y=(A1−A2* X+A3* X ↑ 2−A4 * X ↑ 3 +A5 * X ↑ 4)/

(C1−C2*X +C3* X ↑ 2 − C4*X ↑ 3)

(ى) S=2* (P/Q) Ø (K−1)/ (R−3*T) (1/M)

9. خلاصة :

ذكرت في هذا الفصل الكثير من المبادىء الأساسية التي يحتاج تعلمها المبتديء للاتصال مع الحاسبة الالكترونية باستخدام لغة بيسك. يمكن تلخيص أهم المبادىء بالنقاط التالية:

أولاً : تتكون الحروف الهجائية للغة بيسك من:

(أ) حروف اللغة الانكليزية.

(ب) رموز. الأرقام العشرية.

(جـ) رموز خاصة وعلامات العمليات الرياضية المعروفة.

ثانياً : في لغة بيسك، يكون الاتصال بالحاسبة الالكترونية على شكل برنامج. يتكون البرنامج من عبارات متسلسلة برقم عبارة. تتكون العبارة، بصورة عامة، من:

(أ) رقم العبارة.

(ب) كلمة أساسية أو أكثر.

(جـ) معلمة أو أكثر.

ثالثاً : تتعامل لغة بيسك مع نوعين من البيانات:

(أ) بيانات رقمية: وتكون على أشكال منها أعداد صحيحة

أو أعداد حقيقية أو أعداد أسية .

(ب) بيانات وصفية : وهي تشكيلة من رموز حروف هجاء لغة بيسك . وتتكون من ثوابت مقطعية .

رابعاً : هنالك حدود لخزن البيانات ترافق استخدام لغة بيسك في أي حاسبة الكترونية حيث:

(أ) يمكن خزن مدى محدود من البيانات الرقمية .

(ب) يمكن تمثيل البيانات الرقمية بدقة محدودة .

(جـ) تمثل البيانات الوصفية بطول مقطع محدد مسبقاً .

خامساً : تعبر المتغيرات في لغة بيسك إما عن ثوابت رقمية أو ثوابت مقطعية .

في حالة تعبيرها عن ثوابت رقمية فهنالك المتغيرات البسيطة أو المتغيرات ذات الأبعاد. تسمى المتغيرات البسيطة برمز من حروف الهجاء الانكليزية متبوع برقم واحد اختياري من الأعداد العشرية .

أما المتغيرات ذات الأبعاد، فتسمى برمز من حروف الهجاء الانكليزية متبوع مباشرة بقوسين فيهما إما معلمة واحدة أو معلمتين تفصل بينهما فارزة .

تعبر المتغيرات المقطعية عن الثوابت المقطعية وتميز عن المتغيرات الرقمية بعلامة الدولار توضع مباشرة بعد رمز من حروف الهجاء الانكليزية وبدون وجود رقم عشري . تكون المتغيرات المقطعية إما بسيطة أو ذات أبعاد .

سادساً : تتعامل لغة بيسك مع ثلاثة أنواع من التعابير الرياضية :

(أ) التعابير الجبرية. وتتركب من متغيرات وثوابت ودوال تجمع بينهما عمليات حسابية كالجمع والطرح والقسمة والضرب والرفع لأس. يوجد نظام أسبقية تنفيذ العمليات الحسابية أعلاه إذا تواجدت في تعبير جبري واحد.

(ب) التعابير المقطعية. وتتركب من إما متغير مقطعي أو ثابت مقطعي. توجد عملية وصل تعمل على وضع الثوابت المقطعية متسلسلة.

(جـ) تعابير المتراجحة. وتتركب من تعبيرين جبريين أو تعبيرين مقطعيين يفصل بينهما عملية متراجحة واحدة. وهي ستة عمليات.

سابعاً : برنامج لغة بيسك هو سلسلة من أسطر مرقمة يكون آخرها عبارة نهاية البرنامج وتحتوي كل عبارة على كلمة أساسية أو أكثر. تبتدأ العبارة برقم العبارة والذي يميزها عن سواها ويعتبر عنواناً لها عند تنفيذ البرنامج.

10. كلمات قد تعلمتها :

برنامج	program
رموز	characters
مفردات	vocabulary
قواعد نحو	syntax
حروف هجائية	Alphabet
العبـــارة	statement
كلمة أساسية	key word
معلمة؛ وسيط	parameter
بيانات رقمية	numeric data
بيانات وصفية	textual data
ثوابت رقمية	numeric constants
ثوابت مقطعية	string constants
مقطـــع	string
مـــدى	range
دقـــة	precision
بايت	byte
أرقام مؤثرة	Significant figures
لهجـــة	dialect
دون الحد الأدنى	UNDERFLOW
فوق الحد الأعلى	OVERFLOW

variable	متغير
numeric variable	متغير رقمي
textual variable	متغير وصفي
initialize	تمهـيد
Array variable	متغير ذو أبعاد
mathematical expressions	تعابير رياضية
algebraic expression	تعبير جبري
arithmatic expression	تعبير حسابي
string expression	تعبير مقطعي
conçatenation	عملية وصل
Relational expression	تعبير متراجحة
function	دالـــة

11. تمارين (2) :

استخدم كلمات قد تعلمتها

(أ) متغير رقمي	(و) التعبير الحسابي
(ب) العبارة	(ز) متغير وصفي
(جـ) متغير ذو أبعاد	(حـ) مـدى
(د) المقطع	(ط) التعبير المقطعي
(هـ) دقـة	(ى) البيانات الرقمية

1 ـ تتكون ــــــــ من رقم للتسلسل وكلمـة أساسيـة أو أكثر ومعلمـة أو أكثر.

2 ـ تتعامل لغة بيسك مع ــــــــ بعدة أشكال منها كـأعداد صحيحـة أو حقيقية أو أسية.

3 ـ يتكـون ــــــــ من رمـوز متجـاورة لحـروف هجـاء لغـة بيسـك. ويحصر بين قوسي الاقتباس.

4 ـ تخزن أي خاسبة الكترونيـة ــــــــ عدد من البيانات الرقمية لايجـوز تخطيه.

5 ـ تحدد لهجة لغة بيسك أكبر عدد لـرموز الثـابت الرقمي مـا عدا العـلامة العشرية ويدعى ذلك بـ ـــــــــ خزن البيانات الرقمية.

6 ـ تمثل البيانات الرقمية في برنامج بلغة بيسك باسم يدعى ـــــــــ .

7 ـ يتركب ـــــــــ من متغير مقطعي أو ثابت مقطعي.

8 ـ إن ـــــــــ هـو مجمـوعـة متغيـرات لهـا نفس الاسم ولكن تتميـز باستخدام معلمة واحدة أو أكثر داخل قوسين.

9 ـ هنالك نظام أسبقية لتنفيذ العمليات الحسابية في ـــــــــ .

10 ـ تمثـل الثـوابت المقـطعيـة في بـرنـامـج بلغـة بيسـك بـاسم يـدعى ـــــــــ .

صح أم خطأ:

1 ـ تتـألف حـروف هجـاء لغـة بيسـك من حـروف هجـاء اللغـة الـروسيـة والأرقام الرومانية مع بعض الرموز الخاصة.

2 ـ تعتبر المعلمة وسطاً لتنفيذ العمليـة الريـاضية المقتـرحة من قبـل الكلمة الأساسية في عبارة برنامج البيسك.

3 ـ إن هدف لغة بيسك هو معالجة بيانات رقمية فقط.

4 ـ إن مدى خزن البيـانات المتعـارف عليه من بين لهجـات لغة بيسـك هو (6) أرقام مؤثرة.

5 ـ يمكن أن يـظهـر أي رمــز يستـطاع طبعــه بلوحـة مفــاتيـح الحــاسبـة

الالكترونية ضمن الثابت المقطعي ما عدا علامة الاقتباس.

6 ـ يخزن المقطع (string) في الحاسبة الالكترونية بتخصيص بايت واحد لكل رمز، لذلك فإن للمقطع عدد لا نهائي من الرموز.

7 ـ يعرّف المتغير الرقمي في لغة بيسك برمزين؛ يؤخذ الأول من حروف هجاء اللغة الانكليزية والثاني من الأرقام العشرية.

8 ـ يمكن أن يمثل المتغير ذو الأبعاد كلاً من البيانات الرقمية والبيانات الوصفية في برنامج بلغة بيسك.

9 ـ تبنى التعابير الجبرية في لغة بيسك من عمليتي الجمع والطرح فقط.

10 ـ يتألف برنامج البيسك من أسطر مرتبة برقم العبارة بحيث يكون آخر سطر هو لعبارة نهاية البرنامج.

أجب باختصار :

1 ـ ما هو تركيب البرنامج والعبارة في لغة بيسك؟

2 ـ صممت لغة بيسك لمعالجة نوعين من البيانات. اذكرهما مع الأمثلة.

3 ـ في لغة بيسك، ما هي قواعد كتابة:

(أ) الثابت الرقمي.
(ب) الثابت المقطعي.

4 ـ تستطيع حاسبة الكترونية خزن أعداد حقيقية ضمن المدى ($\pm 0.1\times10^{-38}$) إلى ($\pm 0.99\times10^{+38}$) بدقة قدرها (7) أرقام مؤثرة.

وتسمح بخزن ثابت مقطعي له (20) رمزاً فقط. كيف ستخزن البيانات التالية:

(أ) $(-8765.91 \times 10^{+35})$

(ب) "I LIKE PROGRAMMING IN BASIC"

(جـ) (−9.01 E−8)

(د) (−0.888888889)

(هـ) $(73529.84 \times 10^{+33})$

5 ـ تعرّف على نوع المتغيرات التالية واذكر لكل منها مثال من البيانات التي يمكن أن تمثلها في لغة بيسك.

(أ) E9

(ب) L$

(جـ) C$(7)

(د) H (5)

(هـ) G(91 , 13)

6 ـ كيف تميز، في لغة بيسك، المتغير الوصفي عن المتغير الرقمي سواء أكان بسيطاً أو ذي أبعاد؟

7 ـ لماذا صمم علماء الحاسبة الالكترونية في لغة بيسك نوع من المتغيرات يدعى بالمتغير ذو الأبعاد؟

8 ـ ما هي العمليات والاجراءات التي تمكن لغة بيسك المبرمج من إنجازها باستخدام حاسبة الكترونية؟

9 ـ اذكر العمليات الرياضية التي يمكن أن تنجز في:

(أ) التعابير الجبرية.

(ب) التعابير المقطعية.

(جـ) تعابير المتراجحة.

10 ـ أعد صياغة المعادلة الجبريـة التاليـة بشكل تتقبله لغـة بيسك. ومن ثم نفذ عملياته الحسابية حسب نظام الأسبقية المعروف في لغة بيسك.

$$z = \frac{\sqrt[4]{x \,.\, y}}{\sqrt[3]{v \div w}} + \frac{33\,(y-w)^n}{17\,(v-x)^7} \Big]^{\frac{1}{m}}$$

اختار الجواب الصحيح :

1 ـ إذا كانت دقة خزن حـاسبة الكتـرونية للبيـانات الـرقمية هـو (6) أرقـام مؤثرة. وأدخل إليها عدد مقـداره (−0.00123456) فإنهـا ستخزنـه على أنه ــــــــــ

(أ) (−0.00123)

(ب) (−0.0012345)

(جـ) (−0.001234)

(د) (−0.00123456)

2 ـ تكتب عملية الوصل في التعبير المقطعي بالصيغة التالية ــــــــــ

(أ) X$ * Y$

(ب) X$ ÷ Y$

(جـ) X$ − Y$

(د) X$ + Y$

3 ـ حسب نظام الأسبقية في التعبير الحسابي ــــــــــ

(أ) تنفذ عمليات ما بين القوسين قبل عملية الرفع لأس.

(ب) تنفذ العمليات من اليسار الى اليمين.

(جـ) تنفذ عملية القسمة قبل الضرب.

(د) تنفذ عملية الطرح قبل الجمع.

4 ـ من عمليات تعبير المتراجحة ما يرمز له بـ ــــــــــــ

(أ) ?

(ب) %

(جـ) ↑

(د) > <

5 ـ تكتب المعادلة الجبرية التالية

$$t = \sqrt[n]{7d} \div (d^3 - c)$$

في لغة بيسك بالصيغة التالية:

(أ) T = (7 . D) ↑ (1/M)/D ↑ 3 – C

(ب) T = (7 * D) ↑ (1/M)/D ↑ 3 – C

(جـ) T = (7 * D) ↑ 1/M / (D ↑ 3 – C)

(د) T = (7 * D) ↑ (1/M)/ (D ↑ 3 – C)

6 ـ مثال على ثابت مقطعي يمكن أن يأخذ الشكل التالي ــــــــــــ

(أ) "BASIC"

(ب) 1986

(جـ) "72.35"

(د) 0.9999×10^{-38}

7 ـ يمكـن أن يصــاغ مـتغيــر مقــطعي ذو بعــديـن كمــا فـي المـثــال التالي ــــــــ

(أ) X 7$(K−1 , K)

(ب) X$(K−1 , K)

(جـ) X$7 (K−1 , K)

(د) 7 X$(K−1 , K)

8 ـ يتم تنفيـذ أسطر بـرنامـج البيسك بـالتسلسل ابتـداء من أول سـطر حتى يتحقق :

(أ) طلب تنفيذ عبارة أخرى بواسطة عبارة سيطرة .

(ب) حدوث شرط استثنائي بحيث يؤدي الى توقف تنفيذ البرنامج .

(جـ) تنفيذ عبارة التوقف أو عبارة نهاية البرنامج .

(د) أحد الأمور المذكورة أعلاه .

9 ـ لا تؤثر الفراغات على تنفيذ برنامج البيسك إذا لم تظهر:

(أ) ضمن حروف الكلمة الأساسية للعبارة .

(ب) ما بين رموز رقم العبارة .

(جـ) ضمن علامة المتراجحة ذات الرمزين .

(د)في جميع الأماكن المذكورة أعلاه .

10 ـ يجب أن تسبق كـل لكمـة أسـاسيـة في بـرنـامـج البيسـك ــــــــ واحد أو أكثر .

(أ) برقم عبارة .

(ب) بمعلمة .

(جـ) بتعبير رياضي .

(د) بفراغ .

الفصل الثالث

لنبدأ البرمجة بلغة البيسيك

1. مقدمـــة
2. تخصيص البيانات لمتغيرات في برنامج البيسك
3. إظهار النتائج في لغة بيسك
4. طلب إنهاء تنفيذ برنامج البيسك
5. توضيح برنامج البيسك بالملاحظات
6. برامـج توضيحيـة

   اختبر نفسك
7. خزن المعلومات باستخدام عبارتي (READ + DATA)
8. جعل الحاسبة الالكترونية تسأل المبرمج عن البيانات

   اختبر نفسك
9. دوال رياضية جاهزة من مكتبة لغة بيسك
10. يمكنك تعريف الدالة الرياضية التي تحتاجها في مسألتك
11. كيف تطبع النتائج على شكل تقرير؟

    اختبر نفسك
12. مسائل برمجـة محلولـة

13. خلاصة
14. كلمات قد تعلمتها
15. تمارين (3)

1. مقدمة :

عرفت أن مراحل حل مسألة أكاديمية باستخدام حاسبة الكترونية تتلخص بثلاث مراحل هي؛ كتابة خوارزمية الحل، وكتابة برنامج الحل بلغة برمجة مناسبة، وأخيراً تنفيذ البرنامج على حاسبة الكترونية متوفرة. من الفصل الأول، تعرفت على طرق كتابة خوارزمية الحل. وفي هذا الفصل سنعلمك كيف تبرمج بلغة بيسك وذلك بتعريفك على مفردات (vocabulary) لغة بيسك وقواعد نحوها (syntax) مسندة بأمثلة توضيحية.

عندما أعلن الدكتور كمني (Kemeny) والدكتور (Kurtz) من أمريكا عن تصميم لغة برمجة جديدة أسموها بيسك (BASIC) في جامعة دارموث (Dartmouth) . أكدا أنه يمكن تعلم هذه اللغة في ثلاثة ساعات! وقد أثبتت هذه اللغة منذ عام (1964)أنها سهلة عند التعلم وسهلة عند الاستخدام وسهلة حين التذكر كما أن لها القابلية على تجسيد حل الكثير من المسائل الاكاديمية والتجارية.

بعد دراستك لهذا الفصل سيكون بإمكانك كتابة برامج بسيطة بلغة بيسك. بحيث سيعلمك هذا الفصل الأمور التالية:

1 ـ بعض مفردات لغة بيسك وقواعد نحوها.
2 ـ تغذية التعبير الحسابي في برنامج بيسك.

3 ـ إخراج النتائج مع جمل مفيدة.

4 ـ كيفية إنهاء تنفيذ برنامج البيسك.

5 ـ إيضاح برنامج البيسك بالملاحظات.

6 ـ خزن المعلومات المتوفرة في برنامج البيسك.

7 ـ كيف تجعل الحاسبة الالكترونية تسأل المبرمج عن البيانات؟

8 ـ استخدام دوال رياضية جاهزة من مكتبة لغة بيسك.

9 ـ استخدام دوال رياضية تعرفها أنت.

10 ـ اختيار رقم على نحو عشوائي.

11 ـ كيفية طبع النتائج على شكل تقرير؟

2. تخصيص البيانات لمتغيرات في برنامج بيسك :

يتم تقديم التعابير الحسابية في برنامج بيسك عن طريق الكلمة الأساسية (LET) ، وهي بمعنى (اجعل). حيث تتركب عبارة (LET) من رقم العبارة والكلمة الأساسية (LET) ومن ثم التعبير الحسابي.

مثـال :

رقم العبارة ← 20 ، تعبير حسابي ← X/W + Z * Y

```
20  LET A = X/W + Z * Y
```

كلمة أساسية ← LET

ملاحظات :

(1) يمكن أن يكون المتغير ذو أبعاد.

(2) يمكن أن يأخذ التعبير الحسابي أي درجة من التعقيد ولكن بشرط أن تكون له قيمة رقمية واحدة نهائية تخزن تحت اسم المتغير (A) .

(3) إن علامة المساواة (=) في التعبير الحسابي تعني (استبدله بـ). أي استبدل ما مخزون تحت اسم المتغير (A) بناتج التعبير الحسابي في الجهة اليمنى من المعادلة.

(4) لاحظ الفراغات الموجودة بين الكلمة الأساسية والمتغيرات والعلميات الحسابية بحيث جعلت العبارة مقروءة بوضوح من قبل الآخرين.

مثـال :

```
40    LET B=(X+Y)/(W * Z)
```

في عبارة (LET) رقم (40) أعلاه، يتم استرجاع البيانات الرقمية المخزونة في الذاكرة تحت اسم كل من (X)، (Y)، (N) و(Z) في التعبير الحسابي وتنفذ عليها العمليات الحسابية وحسب نظام الأسبقية المذكور في الفقرة (6) من الفصل الثالث.

ومن ثم تستبدل القيمة الرقمية المخزونة تحت اسم المتغير (B) بناتج ذلك التعبير الحسابي.

من المثير للاهتمام تقبّل لغة بيسك لعبارة (LET) التالية:

```
30    LET I = I + 1
```

حيث أن (I=I+1) يعتبر غير مقبول في الجبر الاعتيادي، لكن قواعد نحو لغة بيسك تفسر عبارة (LET) رقم (30) أعلاه كما يلي:

استرجع الثابت الرقمي الموجود حالياً تحت اسم المتغير (I) ، وأضف إليه واحد. ثم استبدله بقيمة المتغير (I) السابقة. من التطبيقات المهمة لعبارة (LET) رقم (30) اعلاه هو استخدامها كعداد (counter) في انشوطة (loop). عرفت معنى كل من عداد وانشوطة من الفصل الأول، ولكن سيأتي شرحها بالتفصيل في فصل لاحق يخص تكرار الحسابات في لغة بيسك.

يمكن استخدام عبارة (LET) لإدخال البيانات مباشرة الى برنامج بيسك، أي يمكن استخدام عبارة (LET) لخزن البيانات الرقمية أو الوصفية مباشرة الى اسم متغير رقمي أو وصفي موجود في الجهة اليسرى من علامة المساواة.

أمثلة :

معناها في لغة بيسك	عبارة (LET)
اخزن الثابت الرقمي (13) في الذاكرة تحت اسم المتغير الرقمي (C).	10 LET C=13
اخزن الثابت الرقمي (97.23) في العنصر التاسع من اسم التغير الرقمي ذو الأبعاد.	20 LET D(9)=97.23
اخزن الثابت المقطعي الموجود بين علامتين الاقتباس تحت اسم المتغير الوصفي (F$)	30 LET F$="BASIC"
تعيين الثابت المقطعي الموجود بين علامتي الاقتباس الى العنصر الخامس من المتغير الوصفي ذو الأبعاد. بحيث يمكن عنونته في عبارات مقبلة بذكر اسم المتغير فقط.	40 LET H$(5) = "SAMI"

استبـدل مـا مخـزون تحـت اسم المتغير (M) بما مخزون في العنصر التاسع من المتغير ذو الأبعاد.	50 LET M=D(9)

ختـاماً نستـطيـع تلخيص مـا تم ذكـره اعـلاه. حيث تستخـدم عبـارة (LET) لتعيين قيم للمتغيرات عن طريق إمـا التغذيـة المباشـرة في البرنـامج أو كنتيجة لتنفيذ العمليات في تعبير حسابي.

(أ) تخصص عبـارة (LET) قيمـة رقميـة لمتغيـر رقمي أو ثـابت مقـطعي لمتغير مقطعي. تكون البيانات في الجهة اليمنى من عـلامة المسـاواة والمتغيرات في الجهة اليسرى.

(ب) يمكن أن تكون المتغيرات في عبارة (LET) ذات أبعاد.

(جـ) لا تخزن علامة قوسي الاقتباس المحيطة بـالثابت المقـطعي في ذاكرة الحاسبة الالكترونية كجزء من بيانات المتغير المقطعي.

(د) في معظم لهجات لغة بيسك يمكنـك حذف الكلمـة الأساسيـة (LET) من العبارة دون الإخلال بوظيفة العبارة.

(هـ) في المخططات الانسيابية، يستخدم رمز المعالجـة المستطيـل الشكل (▭) ليرمز الى عبارة(LET) في لغة بيسك.

3. إظهار النتائج في لغة بيسك :

تستحصل نتائج حل مسألة من حاسبة الكترونية باستخدام لغة بيسك بواسطة عبارة (PRINT)، وهي بمعنى (اطبع). تتركب عبارة (PRINT) من رقم عبارة والكلمة الأساسية (PRINT) ومعلمة أو أكثر. تتألف المعلمة من متغير أو ثابت أو تعبير حسابي.

أمثلة :

معناها في لغة بيسك	عبترة (PRINT)
تطبع الثابت الرقمي المخزون في الذاكرة تحت اسم المتغير الرقمي (Z) .	10 PRINT Z
تطبع مقطع من الرموز مخزون في الذاكرة تحت اسم المتغير الوصفي (X$) .	30 PRINT X $
تطبع الثابت الرقمي كما هو (1986) .	20 PRINT 1986

تـطبـع مقـطـع الـرمـوز بين قـوسي الاقـتبـاس بـدون طـبـع قـوسـي الاقتباس.	50 PRINT "BASIC"
تـطبع نتيجـة تنفيذ التعبيـر الحسابي بـاستخدام قيم المتغيـرات المتوفـرة وحسب نظام أسبقية تنفيـذ العمليات الحسابية.	40 PRINT X/Y + Z*W

احتوت عبارة (PRINT) في الأمثلة أعـلاه، على معلمة واحـدة فقط. أما إذا كان هنالك أكثر من معلمة فيجب فصلها برمز. وقد اقتـرحت قواعـد نحـو لغة بيسـك. أما الفـارزة (,) أو الشولـة المنقوطـة (;) تؤدي كـل منهمـا وظيفة محـددة في عبارة (PRINT).

في لغـة بيسك، يقسم السـطر الذي تـطبع عليـه النتائـج إلى خمسة منـاطق. وكما هـو معروف، أن سـطر الطبـع يتألـف من (72) مـوقعـاً. لـذا تجد سطر الـطبع قـد قسم الى خمسة منـاطق طبع. تتـألـف كل منـطقة من (15) موقعاً ما عدا المنطقة الأخيرة فإنها ذات (12) موقعاً.

تطبع نتيجة كل معلمة مسبوقـة بفارزة في عبـارة (PRINT) في منطقـة منفصلة، أما تلك التي تسبقها شولة منقوطة فتطبع مبـاشرة بغض النـظر عن موقع منـطقة الـطبع. في قـواعد نحـو لغة بيسـك، تفسر الفـارزة في عبارة (PRINT) «توقف عن الطبـع في المنطقـة التي أنت فيها وابـدأ في المنطقـة التـالية». أمـا تفسيـر الشـولـة فهـو «انسى تقسيم سـطر الـطبـع إلى منـاطق واستمر في الطبع».

أمثـــلة :

عبارة (PRINT)	معناها في لغة بيسك
10 PRINT X$; Y, Z ↑ Z	بعد طبع الثابت المقطعي المخزون تحت اسم المتغير الوصفي (X$) . تفسر الشولة المنقوطة (;) في لغة بيسك «اترك موقع واحد واستمر في طبع محتويات المتغير الرقمي (Y) . بعد ذلك تأمر الفارزة (,) الحاسبة الالكترونية بطبع نتيجة التعبير الحسابي (Z ↑ Z) في أول موقع من المنطقة التالية من سطر الطبع . وإذا كانت الحاسبة الالكترونية تطبع في المنطقة الخامسة وجاءها أمر الفارزة فإنها ستنتقل الى سطر طبع جديد لإكمال طبع النتائج .
30 PRINT X$; Y, 40 PRINT Z$; W	نتيجة لوجود الفارزة بعد آخر متغير في عبارة (PRINT) رقم (30) فإن تأثير العبارتين سيكون كتأثير عبارة (PRINT) التالية : 30 PRINT X$; Y, Z$; W

إذا كانت هنالك شولة منقوطة بدلاً من الفارزة في نهاية عبارة رقم (30) في الحقل الأيسر فسيكون تأثير العبارتين هو كتأثير العبارة التالية: 30 PRINT X$; Y ; Z$; W	50 PRINT X$; Y
إذا لم توجد فارزة أو شولة منقوطة في نهاية أول عبارة (PRINT) فإن الحاسبة الالكترونية ستطبع النتائج المطلوبة في العبارة رقم (60) ومن ثم تنتقل الى سطر طبع جديد لإظهار النتائج التي تطلبها عبارة رقم (70) .	70 PRINT Z$; W
إذا لم تحتوي عبارة (PRINT) على أية معلمة فإن الحسابة الالكترونية لا تطبع شيء. وهي الطريقة في لغة بيسك لترك سطر بدون طبع نتائج عليه.	80 PRINT

من الأمثلة أعلاه، عرفنا تأثير وجود الفارزة أو الشولة المنقوطة بين أكثر من معلمة في عبارة (PRINT). حيث نجدهما يؤثران على اتساع مواقع طبع النتائج المطلوبة بعبارة (PRINT) . الآن سنبين أسلوب إخراج البيانات الرقمية كنتائج مطلوب طبعها بعبارة (PRINT) .

يتألف مجال طبع الثابت الرقمي من موقع لكتابة رموز الاشارة، تليه مواقع لكتابة رموز العدد وبعد ذلك يترك موقع واحد كفراغ. إذا كان العدد موجباً لا تكتب إشارته ويترك بدلها فراغ، أما إذا كان العدد سالباً فتكتب علامة الناقص. يكون العدد إما صحيحاً أو حقيقياً أو أسياً. حتى وإن استخدمت الشولة المنقوطة كفاصلة فسوف يترك على الأقل فراغ بعد العدد ليفصله عما سيطبع بعده.

يمكن أن يطبع الثابت المقطعي في أكثر من منطقة طبع. وإذا جاءت فارزة بعد المتغير الوصفي فإن الحاسبة الالكترونية ستترك المواقع المتبقية من منطقة الطبع الحالية وتنتقل الى منطقة الطبع التالية. وقد بينا ذلك في الأمثلة السابقة عن عبارة (PRINT). أما الأمثلة التالية فستبين تفاصيل إخراج البيانات الرقمية.

مثـــال :

```
30   PRINT Z ; – Z
```

لو كانت قيمة المتغير (Z) هي (987865) . فإن النتائج ستطبع كما يلي :

↑ 987865 ↑ – 987865

حيث تجد موقعاً قد ترك فارغاً (مؤشر بسهم) بين العددين، وأن العلامة الموجبة لم تكتب، بل ترك فراغ، بينما الإشارة السالبة قد كتبت.

مثـــال :

```
20   PRINT 0.00; 98.0; 98.07865
```

إذا كانت دقة خزن البيانات هي ستة أرقام مؤثرة فإن نتيجة العبارة اعلاه ستكون :

↑ 0 ↑ ↑ 98 ↑ ↑ 98.0786

لاحظ أن الرقم (5) قد حذف من آخر عدد لأنه قد تجاوز حدود الدقة. لاحظ الأسهم، حيث تجد السهم يمثل فراغاً.

مثـــال :

```
60   PRINT 987865/ 10 ↑ 10; – 987865/ 10 ↑ 8
```

بما أن النتائج ستكون أصغر من (0.1) بكثير، فستطبع النتائج على شكل أسي، حيث تجد الحرف(E)(يمثل الأساس (10) . ستظهر النتيجة كما يلي :

```
↑ 9.87865 E – 05 ↑  – 9.87865 E – 03
```

مثـــال :

```
70   PRINT 987865 * 10 ↑ 10;
80   PRINT –987865* 10 ↑ 12
```

بما أن النتائج ستكون أعداد أكبر من (10^6) فإنها ستطبع على شكل اسي، كما يلي :

```
↑ 9.87865 E+15  ↑  –9.87865 E+17
```

في المخطط الانسيابي يرمز لإخراج النتائج بمتوازي أضلاع (▱). لذا يمكن ترجمة المطاليب في متوازي الأضلاع باستخدام عبارة (PRINT) في لغة بيسك.

لتلخيص ما تم ذكره، تستخدم عبارة (PRINT) لإظهار النتائج. وهي تتألف من رقم العبارة والكلمة الأساسية (PRINT) ومعلمة أو أكثر. من الملاحظات المهمة حول عبارة (PRINT) :

(أ) يمكن فصل أكثر من معلمة في عبارة (PRINT) بفارزة (,) أو شولة

منقوطة (;) تعمل الفارزة على نقل طبع النتائج الى منطقة طبع جديدة. أما الشولة المنقوطة فتؤيد على استمرار الطبع بدون الانتقال الى منقطة طبع جديدة.

(ب) يتم إحاطة الثابت الوصفي بقوسي الاقتباس في عبارة (PRINT) ولكن عند الطبع يظهر الثابت الوصفي فقط بدون قوسي الاقتباس.

(جـ) يمكن استخدام عبارة (PRINT) لإيجاد قيمة تعبير حسابي. حيث تطبع النتيجة الرقمية بالمواصفات التالية:

(١) إذا كان العدد صحيحاً لا تطبع أصفار بعد العلامة العشرية.
(2) هنالك حدود دقة لا يمكن تجاوزها.
(3) تمثل الأعداد الصغيرة جداً والكبيرة جداً باستخدام الشكل الأسي.
(4) يترك فراغ بدل علامة الزائد في العدد الموجب.
(5) يترك فراغ بعد إكمال طبع العدد إذا كانت الشولة المنقوطة هي الفاصلة عن الحد التالي في عبارة (PRINT).

(د) يتم ترجمة رمز متوازي الأضلاع عند إخراج النتائج في المخطط الانسيابي الى عبارة (PRINT) في لغة بيسك.

4. طلب إنهاء تنفيذ برنامج البيسك :

يتم تنفيذ عبارات برنامج بيسك بالتسلسل ابتداء من أول عبارة حتى تواجه عدة أمور، قد ذكرت في الفقرة (7) من الفصل الثاني، مما يؤدي الى توقف تنفيذ البرنامج. ومن هذه الأمور مواجهة إما عبارة (STOP) أو عبارة نهاية البرنامج (END) .

تتركب كل من عبارة (STOP) أو عبارة (END) من رقم العبارة والكلمة الأساسية (STOP) أو (END) وبدون معلمة. لكن يكون لعبارة (END) أكبر رقم عبارة يواجه في برنامج بيسك، أي تظهر كآخر عبارة في برنامج البيسك. لاحظ البرنامج الموجود في الفقرة (د) والفقرة (7) من الفصل الثاني كأمثلة على موقع العبارتين.

مثـــال :

```
60   STOP
110  END
```

يمكن أن تظهر عبارة(STOP) في عدة أماكن من البرنامج مما يسبب توقف تنفيذ البرنامج. بينما تظهر عبارة (END) كآخر عبارة في برنامج بيسك لأنها ستنهي تنفيذ البرنامج منطقياً في الحاسبة الالكترونية.

يتــرجم الـرمــز الـطرفي البيضـــوي () في التخـطيط الانسيـابي إذا كان المقصـود منه إنهـاء خطوات الحـل بعبـارة (END) في لغة بيسك.

5. توضيح برنامج البيسك بالملاحظات :

من صفات البرنامج الجيد، بصورة عامة، أن يكون سهل القراءة. وذلك عن طريق توضيح الهدف من البرنامج، وتعريف المتغيرات المستخدمة فيه، وتجهيز فواصل مرئية بين العبارات. تضمن لغة بيسك كل الأمور المذكورة اعلاه عن طريق عبارة (REM)، وهي تعني (ملاحظة). تجهّز عبارة (REM) المعلومات فقط للشخص الذي يقرأ البرنامج بدون أن يتم تنفيذها كما يحدث في عبارات لغة بيسك.

تتركب عبارة (REM) من رقم العبارة والكلمة الأساسية (REM) بعدها تكتب أية ملاحظات أو تعلّيق أو مقطع من رموز أو حتى لا يكتب شيء بعدها.

مثـــال :

```
10   REM I LIKE PROGRAMMING IN BASIC
20   REM  *****************
30   REM
```

يمكن أن تظهر عبارة (REM) في أي موقع في برنامج بيسك.

6 برامج توضيحية :

سنضرب أمثلة على شكل برامج لتوضيح عمل العبارات الخمسة (REM ، STOP ، END ، PRINT ، LET) المذكورة في الفقرات السابقة من هذا الفصل.

مسألة برمجة: أوجد معدل درجة درس اللغة الانكليزية لأربعة طلاب :

رياض = 75

سامي = 70

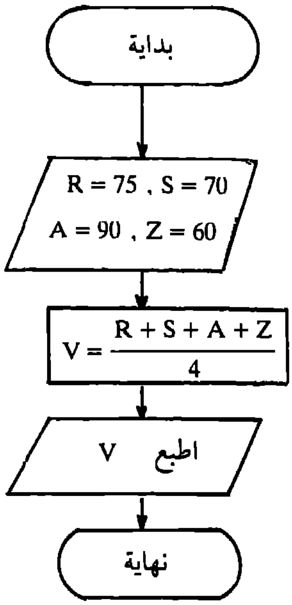
أياد = 85

زكي = 50

استخدم لغة بيسك لكتابة برنامج معدل الدرجات. يمكن إيجاد معدل الدرجات اعلاه، بجمعهم ومن ثم القسمة على (4) .

الحل : لحل مسألة بواسطة البرمجة يجب، أولاً ؛ كتابة خوارزمية الحل، ثانياً؛ كتابة برنامج الحل، ثالثاً تنفيذ البرنامج على حاسبة الكترونية. لذا سنكتب خوارزمية حل المسألة أعلاه وذلك بمعرفة البيانات المتوفرة والنتيجة المطلوبة والصيغة الرياضية التي توصلنا إلى تلك النتيجة. وجميعها متوفر في نص السؤال. سنستخدم المخطط

الانسيابي لتصوير خوارزمية الحل.



1 ـ إدخال البيانات المتوفرة بتحديد اسم متغير لكل ثابت رقمي. تترجم هذه الخطوة بعبارة (LET) في لغة بيسك.

2 ـ معالجة البيانات بتعبير حسابي له غرض إيجاد معدل الدرجات. تترجم بعبارة (LET) .

3 ـ إخراج معدل الدرجات (V). تترجم بعبارة (PRINT) في لغة بيسك.

سنكتب برنامج بيسك مباشرة من المخطط الانسيابي بدون توضيح خطواته بعبارة (REM) :

```
10  LET R=75
20  LET S=70
30  LET A=85
40  LET Z=50
50  LET V= (R+S+ A+Z)/4
60  PRINT "AVERAGE OF GRADE ="; V
70  STOP
80  END
```

الآن يمكن توضيح برنامج البيسك اعلاه باقحام عبارات (REM) بين عبارات برنامج بيسك. ولأننا رقّمنا عبارات البرنامج تصاعدياً وبفجوة قدرها (10) لذا يمكننا اقحام عبارات إضافية إن لزم الأمر.

```
02  REM COMPUTE AVERAGE OF GRADES
04  REM V=AVERAGE OF GRADES
06  REM R, S, A, Z= GRADES FOR STUDENTS
08  REM ****************************************************
10  LET R=75
20  LET S=70
30  LET A=85
40  LET Z=50
50  LET V= (R + S + A + Z)/4
55  REM ****************************************************
60  PRINT "AVERAGE OF GRADES ="; V
70  STOP
80  END
```

ستظهر نتيجة البرنامج اعلاه كما يلي :

```
AVERAGE ↑ OF ↑ GRADES ↑ = ↑ 70
```

ملاحظات حول البرنامج :

(أ) يتركب برنامج بيسك من عبارات مرقمة تصاعدياً. تكون آخر عبارة فيها هي عبارة (END) .

(ب) استخدم عبارة (REM) لتوضيح البرنامج .
استخدم عبارة (LET) لإدخال البيانات ولمعالجة العمليات الحسابية .
استخدم عبارة (PRINT) لاخراج النتائج مع عنوان مناسب لتلك النتائج .

(جـ) يمكن دمج كل من عبارة رقم (50) وعبارة رقم (60) بوضع التعبير الحسابي الموجود في عبارة رقم (50) قبل مقطع الرموز المحاط

بقوسي الاقتباس في عبارة (PRINT) رقم (60) وحذف عبارة رقم (50) من برنامج البيسك اعلاه.

(د) من المستحسن التمهيد للمتغير (V) بتصفيره في عبارة (LET) .

مسألة برمجة : اكتب القيمة الرقمية التي ستخزن تحت اسم كل متغير بعد تنفيذ كل عبارة في برنامج البيسك التالي :

Program	X	Y	Z	W	A1	A2	A
10 LET X=2	2	—	—	—	—	—	—
20 LET Y=4	2	4	—	—	—	—	—
30 LET Z=3	2	4	3	—	—	—	—
40 LET W=1	2	4	3	1	—	—	—
50 LET A1= X/W	2	4	3	1	2	—	—
60 LET A2= Z*Y	2	4	3	1	2	12	—
70 LET A= A1+A2	2	4	3	1	2	12	14
80 PRINT "THE RESULT ="; A							
90 STOP							
100 END							

ستظهر نتيجة البرنامج اعلاه كما يلي :

```
THE RESULT = 14
```

تدعى طريقة تقصي قيم المتغيرات المستخدمة في برنامج بيسك بالتنفيذ الجاف (Dry Run). وتستخدم للتأكد من صلاحية البرنامج .

مسألة برمجة : إن فكرة هذه المسألة مقتبسة من سؤال ورد في ورقة امتحان (Welsh Joint Education Committee) :

اكتب برنامج بلغة بيسك لحساب أجور الهاتف لزبون واحد

باستخدام المعلومات والصيغ الرياضية في المخطط الانسيابي أدناه.
تتكون البيانات الداخلة من:

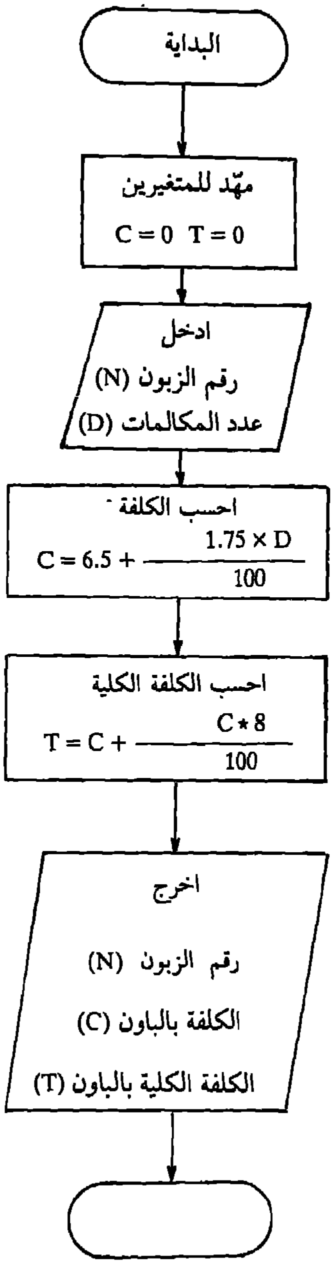
رقم الزبون	CUSTOMER NO. (N)=98765
عدد المكالمات	DIALLED UNITS (D)=1000

أما النتائج المطلوبة فهي ورقة أجور الهاتف مطبوع عليها:

رقم الزبون	CUSTOMER NO. (N)
كلفة المكالمات لهذا الشهر (بالباون)	COST (C)
الكلفة الكلية التي تتضمن الضرائب (بالباون)	TOTAL COST (T)

مطلوب إخراج النتائج اعلاه بالشكل التالي :

CUSTOMER NO	COST (£)	TOTAL COST (£)



1 ـ نمهد المتغيرين بتصفيرهم، لكي نهيء أماكن الذاكرة التي حجزت تحت عنوان اسم المتغير (C) أو (T) .

2 ـ البيانات المتعلقة بالزبون.

3 ـ حساب الكلفة (بالباون).

4 ـ حساب الكلفة الكلية المتضمنة للضرائب (بالباون).

5 ـ إخراج النتائج على شكل قائمة أجور الهاتف تكون لها عناوين (Heading) وكما مطلوب في صيغة السؤال.

الحــل : بمـا أن خوارزميـة الحل قـد أعطيت مسبقـاً ضمن صيغـة السؤال على شكل مخطط انسيابي. فما علينا الآن سوى تحـويل خـطواتها الى عبـارات لغة بيسـك لتكوين بـرنامـج لحل المسـألة. وبـذلك يمكن أن يأخذ البرنامج الشكل التالي :

```
10  REM TELEPHONE BILL
20  REM FOR ONE CUSTOMER
30  REM C= COST (£)
40  REM T= TOTAL COST (£)
50  REM N= CUSTOMER NO.
60  REM D= DIALLED UNITS
70  REM ****************************************************
80  LET C=0
90  LET T=0
100 LET N=98765
110 LET D=1000
120 LET C=6.5 + (1.75 * D)/100
130 LET T= C+ (C * 8)/100
140 PRINT "CUSTOMER NO.", "COST(£)"1 "TOTAL COST(£)"
150 PRINT "?-----------","--------","-----------"
160 PRINT N, C, T
170 STOP
180 END
```

ستظهر النتائج كما يلي :

CUSTOMER NO.	COST (£)	TOTAL COST (£)
----------------------------	------------------	------------------------------
98765	24	25.92

ملاحظات حول البرنامج:

(أ) استخدمت عبارة (REM) :

(1) لتوضيح البرنامج في عبارة رقم (10)و(20) .

(2) لتعـريف متغيرات البـرنـامـج في عبـارة رقم (30)و(40) و(50) و(60) .

(3) لتوفير فاصل مرئي من العلامات المتجاورة عبارة رقم (70) .

(ب) استخدمت عبارة (LET) .

(1) للتمهيد لمتغيرين: عبارة رقم 80) و(90) .

(2) لإدخال البيانات: عبارة رقم (100) و(110) .

(3) لوضع التعابير الحسابية: عبارة رقم (120) و(130) .

(جـ) استخدمت عبارة (PRINT) :

(1) لطبع عناوين للنتائج المطلوبة: عبارة رقم (140) و(150) .

(2) لإخراج النتائج: عبارة رقم (160) .

(د) انهي تنفيذ البرنامج بعبارتي (STOP) و(END) .

(هـ) يتكون البرنامج بمجمله من أربعة مراحل:

(1) تعليقات لجعل البرنامج سهل القراءة بواسطة عبارة (REM) .

(2) إدخال البيانات بواسطة عبارة (LET) .

(3) معالجة البيانات بواسطة عبارة (LET) .

(4) إخراج النتائج بواسطة عبارة (PRINT) .

اختبر نفسك :

1 ـ إذا عـرفنـا أن عبـارة (REM) لا تنفـذ في الحـاسبـة الالكتـرونيـة كبقيـة عبارات لغة بيسك. فلماذا تستخدم تلك العبارة في برنامج البيسك؟

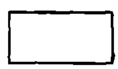
2 ـ ما هي استخدامات عبارة (PRINT) في برنامج بيسك؟ عدد ثلاثة فقط مع ذكر الأمثلة.

3 ـ ما الفرق بين عبارتي (STOP) و(END) ؟

4 ـ اذكر استخدامين لعبارة (LET) في لغة بيسك. وميز أي من التعابير التالية هي صالحة وفق قواعد نحو لغة بيسك، واذكر السبب إذا لم تكن كذلك.

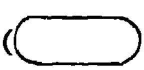
```
10  LET 15=F
20  LET Z$=100
50  LET K=K+1
40  LET A+B=C
60  LET H= "78" +10
70  LET X$= Y$/5
80  LET Z=Z1+Z2/Z3
```

5 ـ كيف تترجم رموز المخطط الانسيابي التالية الى عبارات في لغة بيسك:

(أ) رمز معالجة البيانات ().

(ب) رمز إدخال البيانات ().

(جـ) رمز إخراج النتائج ().

(د) لرمز الطرفي لإنهاء البرنامج ().

7. خزن المعلومات بعبارتي (READ + DATA) :

يعتبر خزن المعلومات واسترجاعها من الوظائف الرئيسية للحاسبة الالكترونية. تنتقل البيانات الى ذاكرة الحاسبة الالكترونية باستخدام عبارات إدخال خاصة في لغة بيسك. وقد تعرفتَ على احداها في الفقرة (2). حيث وجدت أن من استخدامات عبارة (LET) هو إدخال البيانات بتخصيصها الى المتغيرات مباشرة، خاصة إذا كانت البيانات قليلة العدد.

أما إذا أردت إدخال بيانات بكميات أكبر فيمكنك استخدام عبارة (READ)، بمعنى اقرأ. تخصص عبارة (READ) قيم رقمية الى متغيرات. تأخذ قيم البيانات من عبارة (DATA)، بمعنى بيانات.

تتركب عبارة (READ) من رقم العبارة والكلمة الأساسية (READ) ومتغير واحد أو أكثر، تستخدم الفارزة للفصل بين المتغيرات. يمكن أن يكون المتغير رقمي أو وصفي ومن النوع البسيط أو ذو الأبعاد.

وتتركب عبارة (DATA) من رقم العبارة والكلمة الأساسية (DATA) تتبع بسلسلة من البيانات الرقمية أو الوصفية، التي سوف تخصص بالتسلسل من اليسار الى اليمين الى تلك المتغيرات الموجودة في عبارات (READ). تستخدم الفارزة لفصل البيانات عن بعضها.

مثال: سنعيد كتابة برنامج معدل الدرجات الوارد في الفقرة (6) من هذا الفصل بحيث سندخل البيانات بواسطة عبارتي (READ + DATA) بدلاً من الاستخدام المتكرر لعبارة (LET). وبذلك سيأخذ البرنامج الشكل التالي:

```
10  REMCOMPUTE AVERAGE OF GRADES
20  REM V= AVERAGE OF GRADES
30  REM R, S, A, Z = GRADES FOR STUDENTS
40  REM ****************************************************
50  LET V=0
60  READ R, S, A, Z
70  DATA 75, 70, 85, 50
80  LET V= (R+S+A+Z)/4
90  REM ***************************************************
100 PRINT "AVERAGE OF GRADES" =; V
110 STOP
120 END
```

لا توجه عبارة (DATA) أي أمر الى الحاسبة الالكترونية سوى تجهيزها للبيانات التي تحتاجها عبارة (READ) عند التنفيذ. لذا يمكن وضعها في أي موقع من البرنامج حتى قبل عبارة (READ). يمكن استخدام عبارات (DATA) لتجهيز عبارة (READ)، كما يمكن تجهيز أكثر من عبارة (READ) بعبارة (DATA) واحدة.

مثال: سنستخدم نفس المثال أعلاه ولكن بعد تقديم عبارة (DATA) على عبارة (READ) وتجهيز البيانات بعبارتين من عبارات (DATA).

```
10  LET V=0
20  DATA 75, 70
30  READ R, S, A, Z, X$
40  DATA 85, 50, "AVERAGE OF GRADES="
```

```
50  LET V= (R+S+A+Z)/4
60  PRINT X$ ; V
70  STOP
80  END
```

لا بـــد وأنـك قـــد لاحـظت احتـــواء عبـارة (DATA) رقم (40) على بيـانات رقميـة ومقطع من رمـوز محاط بقـوسي الاقتبـاس الـذي سيخصص للمتغير الوصفي (X$) المـوجود في عبـارة (READ). من الضروري التـأكد من مـطابقـة نـوعيـة المتغيـر وعـددهـا في عبـارة (READ) مـع نـوع وعـدد البيانات الموجودة في عبارة (DATA)، وإلا فإن الحاسبة الالكتـرونية سـوف لا تنفذ برنامج البيسك التالي وتظهر رسالة للمبرمج تبين فيها الخطأ.

مثـــال :

```
10  LET V=0
20  DATA 75, 70, 85, 50
30  READ R, S, A, Z, Y
40  LET V= (R+S+A+Z+Y)/5
50  PRINT "AVERAGE OF GRADES ="; V
60  STOP
70  END
```

عند تنفيذ البـرنامـج اعلاه فـإن الحاسبـة الالكترونيـة ستظهـر رسالـة الخطأ التالية:

★ OUT OF DATA AT LINE 30 ★

أي أن هنـالك نقص في البيـانات نـاتجـة عن تنفيـذ عبـارة (READ) رقم (30) ، حيث تجـد المتغيـر (Y) لا تـوجـد لـه بيـانــات في عبــارة (DATA) .

يعتبر إحاطة مقطع الرموز في البيانات الوصفية بقوسي الاقتباس أمراً اختيارياً إذا كان المقطع لا يبدأ بموقع فارغ أو رقم عشري أو علامة (+) أو (-) أو فارزة أو مواقع فراغ في نهاية المقطع. فإذا وجدت تلك الأمور، يجب استخدام قوسي الاقتباس.

مثال: تعتبر عبارات (DATA) التالية صحيحة حسب قواعد نحو لغة بيسك.

```
10  DATA BASIC, GOOD, AUGUST
20  DATA " 28th. MAY", "30th. AUGUST"
30  DATA "+ 123456", PROGRAMMING
```

بينما تعتبر عبارة (DATA) التالية غير صحيحة.

```
40  DATA 28th MAY , 30th AUGUST
```

الآن يمكننا تلخيص المعلومات المتعلقة بعبارتي (READ + DATA). يتم تخصيص البيانات المحتواة في عبارة (DATA) الى متغيرات موجودة في عبارة (READ). وبهذه الطريقة يتم تغذية البيانات لتخزن في الحاسبة الالكترونية ولتعالج فيما بعد بالعمليات الحسابية المناسبة لانجاز هدف البرنامج.

(أ) ويمكن ذكر البيانات في أكثر من عبارة (DATA) وفي مواقع مختلفة من البرنامج لنفس عبارة (READ). والعكس صحيح أيضاً.

(ب) يجب أن تتطابق نوعية البيانات في عبارة (DATA) مع نوعية المتغيرات في عبارة (READ).

(1) يتم إهمال البيانات الفائضة عن عدد المتغيرات.

(2) تظهر رسالة خطأ إذا لم يكن هنالك بيانات كافية في عبارة (DATA) تغطي المتغيرات في عبارة (READ) .

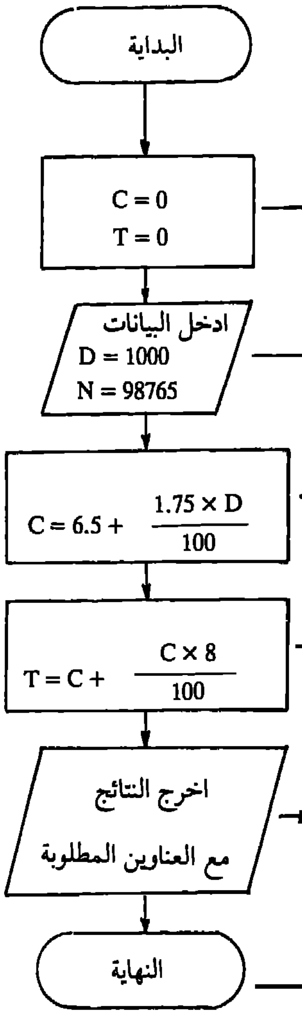
(جـ) لحد الآن يمكنك إدخال البيانات في لغة بيسك بطريقتين:

(1) بواسطة عبارة (LET).

(2)بواسطة عبارتي (READ + DATA) .

(د) في المخطط الانسيابي، يستخدم رمز متوازي الأضلاع لإدخال البيانات. وفي لغة بيسك تنقل المعلومات الموجودة داخل متوازي الأضلاع بواسطة ثنائي عبارتي (READ + DATA) .

مثال: نعود إلى مسألة قائمة أجور الهاتف الواردة في فقرة (6) من هذا الفصل لنحول خوارزمية الحل المصور بمخطط انسيابي الى برنامج بلغة بيسك ولكن باستخدام عبارتي (READ + DATA) بدلاً من عبارة (LET) عند إدخال البيانات.




```
10  LET C=0
20  LET T=0

30  READ N, D
40  DATA 98765, 10000

50  LET C=6.5× (1.75 * D)/100

60  LET T= C+ (C*8)/1000

70  PRINT "CUSTOMER NO.", COST (£)",
80  PRINT "TOTAL COST (£)"
90  PRINT "------------", "---------"
100 PRINT "---------"
110 PRINT N, C, T

120 STOP
130 END
```

تـلاحظ في الجهة اليمنى من المخـطط الانسيابي، عبـارات من لغـة بيسـك تنقل أفكـار ومنطق أسلوب الحـل الى برنـامـج تستـطيـع الحـاسبـة الالكتـرونية تنفيـذه. نرتب العبـارات الموجودة بالشكـل الذي يتـطابق مـع تركيب برنامج بيسك من حيث وجود الفراغات في العبـارة الواحـدة وترتيب العبارات في البرنامج.

يمكن استخـدام قائمـة البيـانـات في عبـارة (DATA) أكثـر من مـرة، بواسطة إيعـاز في لغة بيسـك يدعى عبـارة (RESTORE) أي (أعد خـزن). تتركب عبارة (RESTORE) من رقم العبارة والكلمة الأسـاسية (RESTORE) فقط بدون وجود معلمة. عندمـا تواجـه هذه العبـارة في برنـامج بيسـك فإن عبـارة (READ) التاليـة ستخصص قيم للمتغيرات من بـداية قـائمة البيـانات الموجودة في عبارة (DATA). وبغض النظر من أن قـائمة البيـانات هـذه قد قرأت جميعها يمكن استخدام عبارة (RESTORE) أكثر من مرة في بـرنامـج البيسك.

مثـــال :

```
10   DATA 75, 70, 80, 60, 50, 40, 30, 20, 10
20   READ R, S, A
30   PRINT R; S; A
40   READ R, S, A
50   PRINT R; S; A
60   RESTORE
70   READ R, S, A
80   PRINT R; S; A
90   STOP
100  END
```

ستظهر نتيجة البرنامج أعلاه كما يلي :

75	70	80
60	50	40
75	70	80

ملاحظات :

(1) يخص السطر (20) القيم (75) و(70) و(80) الذي يوفرها سطر (10) الى المتغيرات (R) و(S) و(A) على التعاقب.

(2) يطبع سطر (30) القيم الرقمية المخزونة في المتغيرات.

(3) يخصص سطر (40) القيم (60) و(50) و(40) التي يوفرها سطر (10) الى المتغيرات (R) و(S) و(A) على التعاقب، وتمحى القيم الرقمية القديمة التي تم خزنها بعد تنفيذ سطر (20) .

(4) يطبع سطر (50) القيم الرقمية التي خزنت مؤخراً في المتغيرات.

(5) سطر (60) يعيد تخصيص قيم للمتغيرات المقبلة من بداية قائمة البيانات الموجودة في سطر (10) .

(6) يخصص سطر (70) القيم (75) و(70) و(80) الى المتغيرات (R) و(S) و(A) على التعاقب.

(7) يطبع سطر (80) القيم الرقمية التي خزنت مؤخراً في المتغيرات.

(8) لم تخصص القيم الرقمية (30) و(20) و(10) لأية متغيرات.

تعيد عبارة (RESTORE) الحاسبة الالكترونية الى بداية صف

البيانات الموجودة في عبارات (DATA). بحيث عندما تنفذ الحاسبة الالكترونية عبارة (READ) فإنها تخصص قيم للمتغيرات من مقدمة صف البيانات وتترك الباقي منها لعبارات (READ) لاحقة. يبقى صف البيانات التي تجهزها عبارة (DATA) مخزوناً كما هو بعد القراءة.

8. جعل الحاسبة الالكترونية تسأل المبرمج عن البيانات

عرفت، لحد الآن، أن هنالك طريقتين في لغة بيسك لتغذية البيانات الى الحاسبة الالكترونية. إما بواسطة عبارة (LET) أو بعبارتي (READ + DATA) ، اعتماداً على التطبيق. فيمكن استخدام عبارة (LET) لتخصيص قيمة أولية لمتغير أو عندما تكون البيانات الداخلة قليلة. وتستخدم عبارتي (READ + DATA) عندما يراد تغذية بيانات كثيرة ولا تبدل كثيراً.

هنالك طريقة ثالثة لتغذية البيانات هي بواسطة عبارة (INPUT) أي (ادخل). حيث نجد فائدتها تتجلى :

(1) في التطبيقات التي تتطلب التفاعل بين المبرمج والحاسبة الالكترونية مثل الألعاب أو محاكاة (Simulation) بعض الفعاليات أو برامج اختبار الطلاب.

(2) عند تكرار تنفيذ صيغة رياضية مع مجموعة مختلفة من البيانات. فمن غير العملي أن نعيد البرنامج في كل مرة نبدل البيانات. بل تسمح لغة بيسك بإدخال البيانات أثناء تنفيذ البرنامج بواسطة عبارة (INPUT) .

تتركب عبارة (INPUT) من رقم العبارة والكلمة الأساسية (INPUT) ومتغير واحد أو أكثر. تفصل الفارزة بين المتغيرات. وتكون المتغيرات رقمية أو وصفية، وبسيطة أو ذات أبعاد.

تعمل عبارة (INPUT) مثل عبارة (READ) ولكن بدون مرافقة عبارة (DATA) . بحيث عندما تنفذ الحاسبة الالكترونية عبارة (INPUT)، يظهر طلب لطبع البيانات على الجهاز الذي أدخل بواسطته برنامج بيسك. يكون طلب إدخال البيانات إما على شكل صوت موسيقي ينبه المستفيد أو إظهار علامة استفهام (?) أو سهم (←) أو رمز أكبر من (>) على جهاز إدخال البيانات.

أمثلة :

عبارة (INPUT)	كيفية إدخال البيانات
50 INPUT D, E, F	? 1.1, 2.2, 3.3
30 INPUT X$, H$	? "BASIC", "PROGRAMMING"
60 INPUT E, F, X$	? 2.2, 3.3 "BASIC"
80 INPUT I,J, Z (I, J)	? 1, 2, 97.28

ملاحظات :

(1) تم فصل البيانات الداخلة بواسطة فارزة. وبعض لهجات لغة بيسك يستخدم الفراغ كفاصل بين البيانات.

(2) يمكن تغذية بيانات رقمية ووصفية في عبارة (INPUT) واحدة.

(3) يمكن إدخال البيانات الوصفية بدون قوسي الاقتباس إذا لم يبتدأ مقطع الرموز برقم عشري أو علامة (+) أو (−) أو فارزة.

(4) يجب مطابقة نوعية البيانات المدخلة مع نوعية المتغيرات في عبارة (INPUT) .

(5) يتم إدخال قيم المتغيرات ذات الأبعاد من اليسار الى اليمين.

وعند تنفيذ عبارة (INPUT)، يتوقف تنفيذ البرنامج وتنتظر الحاسبة الالكترونية إدخال البيانات بعد العلامة التي تظهرها. لا تستمر الحاسبة الالكترونية في تنفيذ البرنامج حتى يتم إدخال البيانات المناسبة للمتغيرات الموجودة في عبارة (INPUT) .

مثال: سنستخدم عبارة (INPUT) لتغذية بيانات برنامج إيجاد معدل درجات الطلاب الأربعة في درس اللغة الانكليزية المذكور في الفقرتين (6) و(7) من هذا الفصل.

```
10  LET V=0
20  INPUT R, S, A, Z
30  LET V= (R+S+A+Z)/4
40  PRINT "AVERAGE OF GRADES ="; V
50  STOP
60  END
```

عند تنفيذ البرنامج أعلاه ستظهر علامة استفهام على نفس الجهاز الذي أدخلت منه البرنامج فتجيب أنت بطبع أربعة بيانات رقمية. كما يلي:

```
? 75, 70, 85, 50
AVERAGE OF GRADES = 70
```

من السهل نسيان ما هي المتغيرات أو نوعيتها عند ظهور علامة

الاستفهـام المفاجىء، خصـوصـاً في البـرامـج الكبيـرة أو بعـد مـرور زمن طويل لآخر مرة نفـذ بها هـذا البرنـامج. لـذلك فمن الأفضـل وضع عبـارة (PRINT) قبـل عبارة (INPUT) يـذكر فيهـا تعليقات منـاسبـة، تبين فيهـا مـا المطلوب كتابتـه عند ظهـور علامـة الاستفهام. سنتـرك لك محـاولة تـطبيق المـلاحظة على المثـال أعلاه. ونحن سنـطبقها على مسـألة أجـور الهـاتف الموجودة في الفقرة (6) و(7) من هذا الفصل.

مثـــال :

```
10  REM TELEPHONE BILL
20  LET C=0
30  LET T=0
40  PRINT "CUSTOMER NO. (N), DIALLED UNITS (D)"
50  INPUT N, D
60  LET C=6.5+ (1.75 * D)/100
70  LET T=C+ (C * 8)/100
80  PRINT "CUSTOMER NO.", "COST (£)", "TOTAL COST(£)"
90  PRINT "-----------", "---------", "-----------"
100 PRINT N, C, T
110 STOP
120 END
```

سيظهر عند تنفيذ البرنامـج اعلاه، تعليقـاً يبين ما الـذي ستدخله من بيـانات ثم تـظهر عـلامة استفهـام لتطبـع أنت بعدهـا قيمة تلك البيـانـات، بعدها تخرج الحاسبة الالكترونية النتائج المطلوبة كما يلي :

```
CUSTOMER NO. (N), DIALLED UNITS (D)
? 9876,5, 1000

CUSTOMER NO.          COST(£)          TOTAL COST(£)
------------------    ------------     ------------------
98765                 24               25.92
```

أصبح البرنامج أعلاه أكثر مرونة بعد إدخال عبارة (INPUT). حيث نستطيع الآن تطبيق البرنامج على أي زبون بمجرد طبع رقمه وعدد مكالماته الشهرية عند التنفيذ.

الآن ماذا سيحدث لو استجاب المبرمج أو المستفيد على نحو غير صحيح لطلب الحاسبة الالكترونية بتغذية البيانات؟ طبعاً، لن تتقبل ذلك، وستظهر الحاسبة الالكترونية عدم رضاها برسالة خطأ تبين فيها موضع الخطأ المرتكب ومن أشهر الأخطاء سنذكر الحالات التالية:

(أ) إذا كانت البيانات أكثر من عدد المتغيرات في عبارة (INPUT) فإن الحاسبة الالكترونية ستطلب إعادة طبع البيانات مرة أخرى كما يلي:

```
20   INPUT X, Y, Z
```

فإذا طبعت البيانات التالية كاستجابة للعبارة رقم (20) اعلاه فستظهر لك رسالة خطأ:

```
? 90, 80, 70, 60, 50, 40
TOOMUCHFATA;REYTYPE LINE
?
```

أي أن «عدد البيانات أكبر من عدد المتغيرات لذا أعد طباعة السطر». فتعيد كتابة السطر عند علامة الاستفهام في السطر الذي يلي رسالة الخطأ. وفي بعض لهجات البيسك، تهمل البيانات الفائضة، وتأخذ بنظر الاعتبار البيانات التي تفي بالغرض.

(ب) إذا كانت البيانات أقل من المطلوب تغذيته لكي يخصص للمتغيرات

الموجودة في عبارة (INPUT). فلو صادفت الحاسبة الالكترونية العبارة التالية :

```
40   INPUT G, H, I, J, K
```

وأظهرت الحاسبة الالكترونية علامة استفهام واستجبت أنت بإدخال ثلاثة قيم رقمية فقط. فستظهر لك رسالة خطأ كما يلي :

```
? 60,30, 60
NOT ENOUGH DATA ;RETYPE LINE
?
```

أي «ليست البيانات كافية ؛ أعد طباعة السطر». فتعيد كتابة السطر بالبيانات الكافية عند علامة الاستفهام الجديدة.

(جـ) إذا لم يكن هنالك تطابق بين نوع المتغيرات في عبارة (INPUT) مع نوعية البيانات المطبوعة بعد علامة الاستفهام.

```
70   INPUT D, X$ , F
```

واستجبت بعد ظهور علامة الاستفهام بإدخال البيانات المطبوعة أدناه، فإن الحاسبة الالكترونية ستظهر لك رسالة خطأ كما يلي :

```
? "BASIC", 100, "PROGRAMMING"
.WRONG TYPE OF DATA; RETYPE LINE
?
```

أي «تم تخصيص نوعية بيانات غير صحيحة ؛ أعد طباعة السطر». فتطبع أنت بيانات أخرى تطابق نوعية المتغيرات في عبارة (INPUT) أي تخصيص بيانات رقمية للمتغيرات الرقمية وبيانات وصفية للمتغيرات الوصفية.

يمكننا الآن تلخيص المعلومات المتعلقة بعبارة (INPUT). حيث تسمح هذه العبارة للمستفيد أو المبرمج من إدخال البيانات أثناء تنفيذ البرنامج، ولهذا تطبيقات مثيرة منها في الألعاب ومحاكاة بعض الفعاليات مثل قيادة السيارة أو اختبار الطلاب باستخدام الحاسبة الالكترونية.

(أ) تعمل عبارة (INPUT) مثل عبارة (READ) ولكن بدون وجود عبارة (DATA) حيث يتم إدخال البيانات بواسطة الجهة المستفيدة أثناء تنفيذ البرنامج، وبعد ظهور إشارة متفق عليها على جهاز إدخال البيانات.

(ب) يكون طلب إدخال البيانات إما بواسطة صوت موسيقي ينبه المستفيد من البرنامج أو ظهور علامة استفهام (?) أو سهم (←) أو رمز أكبر من (>) .

(جـ) يمكن أن تطلب عبارة (INPUT) تخصيص بيانات لأكثر من متغير. تفصل الفارزة بين المتغيرات.

(د) يمكن إظهار جملة توضيحية قبل إشارة إدخال البيانات، بوضعها في عبارة (PRINT) يكون موقعها قبل عبارة (INPUT) .

(هـ) يجب مطابقة نوع البيانات المدخلة مع نوع المتغيرات في عبارة (INPUT) .

(و) في المخطط الانسيابي يكون رمز إدخال البيانات على شكل متوازي أضلاع. وفي لغة بيسك، تغذى البيانات المتوفرة في رمز متوازي الأضلاع الى الحاسبة الالكترونية بثلاثة طرق:

(1) بواسطة عبارة (LET) .

(2) بواسطة عبارتي (READ + DATA) .

(3) بواسطة عبارة (INPUT) .

اختبر نفسك :

1 ـ هنالك ثلاثة عبارات في لغة بيسك تستخدم لإدخال البيانات. اذكرها، وبيّن أي منها يجب استخدامه في الحالات التالية :

(أ) إذا أردت الحاسبة الالكترونية هي تسأل الجهة المستفيدة عن البيانات.

(ب) إذا كانت البيانات قليلة العدد.

(جـ) إذا كانت البيانات كثيرة العدد وغير قابلة للتبديل.

(د) إذا كانت البيانات تتغير في كل مرة ينفذ فيها البرنامج .

2 ـ في لغة بيسك أي العبارات تستخدم لنقل المعلومات من الرمز المتوازي الأضلاع؟

3 ـ اكتب عبارة (READ) واحد لتخصيص البيانات الموجودة في عبارات (DATA) التالية الى متغيرات مناسبة :

```
30  DATA "TODAY", 9.5, "31th AUGUST"
40  DATA 7.8, "ZEKI", 1.3
50  DATA 4.5, 6.7, "BASIC"
```

4 ـ كيف ستكون استجابة الحاسبة الالكترونية إذا لم تحسن الجهة المستفيدة الرد على طلبها بتزويدها بالبيانات غير المناسبة. حيث ارتكبت الجهة المستفيدة الأخطاء التالية :

(أ) بيانات أكثر من المتغيرات في عبارة (INPUT) .

(ب) بيانات أقل من المتغيرات في عبارة (INPUT) .

(جـ) عدم تجانس نوع البيانات مع نوع المتغيرات في عبارة (INPUT) .

5 ـ فيما يلي سنذكر أمور يمكن أن تظهر مع عبارة (READ) أو عبارة (DATA) أو عبارة (INPUT)، ويمكن أن تظهر بأكثر من واحدة أو لا تظهر بأي منها، خصص كل منها الى العبارة التي تظهر فيها:

(أ) الشولة المنقوطة (;) .

(ب) الفارزو.

(جـ) قيم رقمية.

(د) متغيرات وصفية.

(هـ) قيم وصفية.

(و) متغيرات رقمية

(ز) مقطع رموز بدون قوسي الاقتباس.

(ح) مقطع رموز محاط بقوسي الاقتباس.

(ط) رقم عبارة.

(ى) كلمة أساسية.

9. دوال رياضية من مكتبة لغة بيسك :

تستطيع لغة بيسك من معالجة البيانات الداخلة بتعابير رياضية ذات عمليات حسابية (+، -، *، /، ↑) أو عمليات مقارنة (=، <، = >، < >، . . .) أو عمليات منطقية (AND , OR) وحتى عمليات وصفية مثل عملية الوصل بين المقاطع الوصفية. عرفت ذلك من الفصل الخاص بأساسيات لغة بيسك. الآن ستعرف أن للغة بيسك القابلية أيضاً على تجهيز دوال رياضية مخزونة في ذاكرتها تدعى بدوال مكتبة لغة بيسك (Library Function)، تستخدم في معالجة البيانات. حيث تم برمجة الحاسبة الالكترونية لايجاد قيم بعض الدوال الشائعة الاستعمال في الرياضيات فقط بذكر اسمها الذي اصطلح عليه، مما يزيد من سهولة استخدام لغة بيسك في حل المسائل الرياضية.

إن الدالة الرياضية هي تعبير رياضي يمثل إجراءات معرفة والتي يمكن تكرار استخدامها في برنامج بيسك بالإشارة فقط الى الاسم الذي اصطلح عليه. تقدم لهجات لغة بيسك العديد من الدوال ولكن جميعها يجب أن تحتوي على الأحد عشر دالة المذكورة أدناه من مكتبة لغة بيسك.

صيغة الدالة في لغة بيسك	معنى الدالة رياضياً
SIN(X)	1 ـ دالة الجيب في المثلثات، × (بالزاوية الشعاعية)
COS(X)	2 ـ دالة الجيب تمام في المثلثات، × (بالزاوية الشعاعية)
TAN(X)	3 ـ دالة الظل في المثلثات، × (بالزاوية الشعاعية)
ATN(X)	4 ـ دالة عكس الظل في المثلثات، × (بالزاوية الشعاعية)
LOG(X)	5 ـ دالة اللوغارتم الطبيعي
EXP(X)	6 ـ دالة اسية، الأساس هو (e) والأس هو (x)
SQR(X)	7 - الجذر التربيعي للحجة (X)
INT(X)	8 ـ أكبر رقم صحيح أصغر من أو مساوي الى (X)
SGN(X)	9 ـ إشارة الحجة (X)
ABS(X)	10 ـ القيمة المطلقة للحجة (X)
RAN	11 ـ دالة تختار رقماً عشوائياً من الفترة [0، 1[

تعامل دوال مكتبة لغة بيسك في التعبير الرياضي مثلما يعامل المتغير الرقمي. حيث يمكنك كتابة أي دالة من الدوال الأحد عشر اعلاه بدل المتغير الرقمي في التعبير الرياضي أو أي عبارة في لغة بيسك. ولكن لا يجوز ترك فراغات بين رموز اسم الدالة أو بين اسم الدالة وقوس الحجة (argument) .

يمكن تقسيم دوال مكتبة لغة بيسك الى ثلاثة فئات حسب الوظيفة التي تؤديها رياضياً.

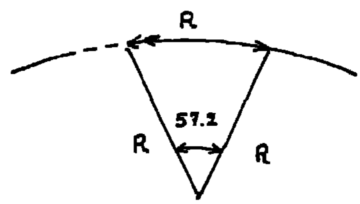
أولاً ـ الدوال المثلثية (TRIGONOMETRIC FUNCTIONS) :

تعتبر الدوال الأربعة الأولى من الجدول اعلاه هي دوال مثلثية. وقد اطلق عليها هذا الاسم لارتباطها بقياس زوايا وأضلاع المثلث في الهندسة الاقليدية. تستخدم الحاسبة الالكترونية الزاوية الشعاعية (Radian)

لقياس الزوايا بينما يفضل البشر القيـاس بالـدرجات (Degree). وفيمـا يلي العلاقة بين القياسين.

$$radian = \frac{180^\circ}{\pi} = 57.295\ degree$$

$$degree = \frac{\pi}{180^\circ} = 0.01745\ radian$$



حيث أن (π) هي النسبة الثابتة

$$\pi = \frac{22}{7} = 3.14159$$

سنرمز للنسبة الثابتة بالمتغير الرقمي (P) :

```
20  LET P= 3.14159
```

يمكنك استخدام الـدالة المثلثيـة بذكـر اسمها فقط مـع ذكر الحجـة بالزاوية الشعاعية. لذا سنعرفك بـالدوال المثلثيـة كلها من خـلال البرنـامج البسيط التالي :

```
10  REM TRIGONOMETRIC FUNCTIONS
20  LET P=3.14159
30  PRINT "SIN(X)"; SIN (-P/2); SIN(0); SIN (P/2)
40  PRINT "COS(X)"; COS (-P/2); COS(0); COS(P/2)
50  PRINT "TAN(X)"; TAN (-P/4); TAN(0); TAN (P/4)
60  PRINT "ANT(X)"; ATN (-1); ATN(0); ATN(1)
70  STOP
80  END
```

ستظهر الحاسبة الالكترونية نتائج البرنامج اعلاه كما يلي :

SIN(X)	−1	0	1
COS(X)	0	1	0
TAN(X)	−1	0	1
ATN(X)	−0.7854	0	0.7854

ثانياً ـ الدوال الأسية (EXPONENTIATION FUNCTIONS) :

تعمـــل الـدوال الأسيـــة على رفـع عـــدد معين الى أس معين. وهي ثلاثة دوال. منها يوجد الجذر التربيعي للحجة (X) .

$$SQR(x) = \sqrt{X} = \uparrow (1/2) = X \uparrow 0.5$$

تعـطي دالة الجـذر التربيعي نتـائج فقط إذا كـانت الحجة مـوجبـة أو مساوية للصفر، أما إذا كانت سالبة فإن الحاسبة الالكتـرونية ستـظهر رسـالة خطأ الى الجهة المستفيدة.

هنالك دالة أسية أخرى توجد اللوغارتم الطبيعي للحجة (X) .

$$LOG(X) = \log_e (X)$$

تـظهر الحـاسبة الالكتـرونية نتيجـة دالة اللوغـارتم الـطبيعي للحجـة الموجبة، أمـا إذا كانت سـالبة أو مسـاوية للصفـر فإن الحـاسبة الالكتـرونية ستظهر رسالة خطأ.

أما الدالة الأسية الثـالثة فهي تعمـل على رفع القيمـة الرقميـة للحجة للأساس (e) :

$$EXP(X) = e^X$$

حيث أن

$$e = 2.71828$$

تستخدم الدوال الأسية الثلاثة اعلاه في عبارات برنامج بيسك كما تستخدم المتغيرات الرقمية. سنوضح ذلك من خلال البرنامج البسيط التالي :

```
10  REM EXPONENTIAL FUNCTIONS
20  PRINT "SQR(X)"; SQR(25); SQR(0)
30  PRINT "LOG(X)"; LOG(1); LOG(10)
40  PRINT "EXP(X)"; EXP(0); EXP(1)
50  STOP
60  END
```

ستكون نتائج البرنامج اعلاه كما يلي :

```
SQR(X)  5    0
LOG(X)  0    1
EXP(X)  1    2.71828
```

ثالثاً ـ الدوال الرياضية (MATHEMATICAL FUNCTIONS) :

بقيت في مكتبة لغة بيسك أربعة دوال تدعى بالدوال الرياضية.

(أ) الدالة التي توجد إشارة الحجة (X) .

$$\begin{aligned} SNG(X) &= +1 \quad \text{if } x > 0 \\ &= \ \ 0 \quad \text{if } X = 0 \\ &= -1 \quad \text{if } X < 0 \end{aligned}$$

(ب) الدالة التي توجد القيمة المطلقة للحجة (X) .

$$\begin{aligned} ABD(X) &= +X \quad \text{if } X \geqslant 0 \\ &= -X \quad \text{if } X < 0 \end{aligned}$$

(جـ) الـدالة التي تـوجد أعلى عـدد صحيح والـذي يعتبـر أقـل أو مسـاوي للحجة (X) .

X	INT(X)
4	4
7.3	7
99.9	99
−9.2	−10
−2.9	−3
−7.3	−8
0.821	0

سنوضح عمل الدوال الثلاثة اعلاه بالبرنامج البسيط التالي :

```
10 REM EXAMPLE ABOUT SGN(X), ABS(X), INT(X)
20 PRINT "SGN(X)"; SGN(9.9); SGN(−0.5); SGN(0)
30 PRINT "ABS(X)"; ABS(9.9); ABS(0); ABS)(−0,5)
40 PRINT "INT(X)"; INT(9.9); INT(−9.9); INT(9)
50 STOP
60 END
```

ستظهر نتائج البرنامج اعلاه كما يلي :

```
SGN(X)  1    0    −1
ABS(X)  9.9  0    0.5
INT(X)  9   −10   9
```

(د)الـدالـة التي تختـار العـدد التـالي من سلسلة اعـداد معـرفـة مسبقـاً ومخزونة في ذاكرة الحاسبـة الالكترونيـة يكون ذلـك العدد في المـدى بين الصفر (0.000000) والواحد (0.999999) . ولهذه الـدالة تـطبيقات مثيرة في الألعاب وبرامج محاكاة الفعاليات المختلفة .

مثــال :

```
10 PRINT RND; RND; RND
```

يمكن أن تظهر الحاسبة الالكترونية الأعداد التالية :

0.350831 0.615274 0.911251

لـو نفذت عبـارة (PRINT) اعلاه مـرة أخرى فسـوف لن تـظهـر نفس الأعداد الثلاثة المذكورة اعلاه.

مثــال : ماذا تنجز العبارة التالية :

```
20 LET N=INT (10 * RND) +1
```

تختار عبارة (LET) اعلاه رقم حظك حيث أنها ستعمل كما يلي :

(1) تولد دالة (LET) عدداً من المدى (0.000000 – 0.999999) .

(2) يـولّـد التعبيـر الحســابي (RND * 10) عـدداً بين الصفــر والعــدد (9.999999) .

(3) تـوجد الـدالة (INT) العـدد الصحيح من العـدد الذي أنتـج من الخطوة (2) . أي سيكون العدد بين الصفر والتسعة.

(4) وبعـد إضافـة واحد، فسينتـج رقم بين الواحـد والعشر وهـو رقم حظك لهـذا الأسبوع قـد اختارتـه لك الحـاسبة الالكتـرونية اعتبـاطيـاً وبـدون تحيز.

10. يمكنك تعريف الدالة الرياضية التي تحتاجها في مسألتك:

تسمح لك لغة بيسك بتعريف الدالة الرياضية التي تحتاجها، والتي ليست ضمن دوال مكتبة لغة بيسك. تستخدم عبارة (DEF) لتعريف الدالة الرياضية. والكلمة (DEF) هي مختصر للكلمة (DEFINE) أي (عرّف). يصاغ اسم الدالة الرياضية بابتدائه بالحرفين (FN) وهما مقتبسان من الكلمة (FUNCTION) أي (دالة). يلي (FN) حرف مناسب من حروف هجاء اللغة الانكليزية، والذي يتبع بقوسين بداخلهما حجة. وبذلك يمكنك تعريف (26) دالة رياضية باستخدام عبارة (DEF) لكل دالة في برنامج بيسك.

تتركب عبارة (DEF) من رقم العبارة والكلمة الأساسية (DEF) تليها اسم الدالة.

مثـــال :

```
10   DEF FNC(X) = 6.5 + (1.75*X)/100
```

تأمر عبارة (DEF) أعلاه، الحاسبة الالكترونية بأن:

(1) يجب أن تكون قيمة الحجة (X) معلومة.

(2) تضرب قيمة الحجة (X) في (1.75) .

)3) تقسم الناتج على (100) .

(4) تضيف الناتج إلى (6.5)

(5) تخزن النتيجة تحت اسم الدالة ( )FN في ذاكرة الحاسبة الالكترونية.

سنضع عبارة (DEF) كجزء من برنامج قائمة أجور الهاتف الذي عهدته كمثال في فقرات هذا الفصل. ولكن لاحظ يجب أن يكون لعبارة (DEF) رقم عبارة أصغر من رقم العبارة التي تستخدم الدالة المعرّفة في هذا البرنامج.

```
 10  REM TELEPHONE BILL
 20  LET D=0
 30  LET T=0
 40  DEF FNC(D) = 6.5 + (1.75 * D)/100
 50  PRINT "CUSTOMER NO. (N), DIALLED UNITS (D)"
 60  PRINT N, D
 70  LET T= FNC(D) + (FNC(D) * 8)/100
 80  PRINT "CUSTOMER NO.", "COST(£)", "TOTAL COST(£)"
 90  PRINT "------------", "---------", "-----------"
100  PRINT N, FNC(D), T
110  STOP
120  END
```

نفّذ هذا البرنامج باستخدام الحاسبة الالكترونية المتوفرة لديك وتأكد من أن له نفس النتيجة التي عهدتها من الفقرات السابقة.

مما تقدم عرفت أنه بإمكان المبرمج استخدام عبارة (DEF) لتعريف الدوال الرياضية التي لا تتواجد في مكتبة لغة بيسك. يمكن الرجوع الى الدالة الرياضية المعرفة بذكر اسمها فقط. واسم الدالة يتكون من الرمزين (FN) متبوع بحرف مناسب من حروف هجاء اللغة الانكليزية، ثم توضع قوسين في داخلهما حجة للدالة الرياضية.

11. كيف تطبع نتائج برنامج بيسك على شكل تقرير ؟

تـطبع نتـائج بـرنـامـج البيسـك بـاستخـدام عبـارة واحـدة هي عبـارة (PRINT). يتم الفصـل بين متغيرات عبـارة (PRINT) باستخـدام الفـارزة أو الشولة المنقوطة. تسبب الفارزة طبع النتائج في مناطق طبع معـرّفة مسبقـاً. وتسبب الشـولة المنقـوطة على بـدء طبع النتـائج في مـوقع الـطبع الـلاحق لآخر نتيجة. يمكنك استخدام كلاهما بسهـولة ولكن على حسـاب المرونـة في طبـع نتائـج البـرنـامـج. ولاكتسـاب أكبـر قـدر من المـرونـة، يمكنـك استخـدام دالـة الجـدولـة (TABULATION FUNCTION) ، والتي تـدعى اختصـاراً دالـة (TAB). تستخـدم دالـة (TAB) فقط في عبـارة (PRINT) للسيطرة على طبع النتائج في مواقع طبع محددة بالضبط.

تتـركب دالة (TAB) من اسم الـدالة (TAB) متبـوع بقوسين يكتب في داخلهمـا حجـة، يمكن أن تكـون ثـابتـاً رقميـاً أو متغيـراً رقميـاً أو تعبيـراً حسابياً.

أمثـــلة :

```
TAB (Z)
TAB (Z/2)
TAB (Z * Z + 5)
TAB (10)
TAB (9 . 8)
```

بحيث تجعل موقع طبع النتائج يبدأ من الثـابت الرقمي الـذي يستحصل من قيمة الحجة. فـإذا كان نـاتج التعبيـر الحسابي عـدداً حقيقياً فـإن الحاسبـة الالكترونية ستـأخذ بنـظر الاعتبار الجـزء الصحيح فقط من العـدد الحقيقي وتترك الجزء العشري.

يتكون سطر الطبع من (72) مـوقعاً للطبـع. تبدأ من المـوقع (صفـر) وتنتهي بالموقع (71). لذا فـإن TAB(11) يعني بدايـة الطبـع من اليسار إلى اليمين في الموقع رقم (2) من سطر الطبع.

مثـــال :

```
10  REM EXAMPLE ABOUT TAB(X)
20  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30  PRINT TAB(11); "L"; TAB(24); "Y"
40  PRINT TAB(15); "P"; TAB(20); "T"
50  STOP
60  END
```

ستظهر نتائج البرنامج اعلاه كما يلي :

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
           L   P    T   Y
```

ملاحظـات :

(1) دائمـاً يحسب الثـابت الـرقمي في دالـة (TAB) من مـوقـع الـطبـع رقم (صفر) سواءً أكان هنالك دالة واحدة أو أكثر من (TAB) .

(2) يجب أن يكـون للثابت الـرقمي في دالة (TAB) أو النـاتـج عن حســاب التعبير الحسابي في دالـة (TAB) اكبر قيمـة لآخر دالـة (TAB) في عبارة (PRINT)، أي يجب أن لا يكـون نهايـة الطبـع في المـوقـع (50) وتأتي

دالة (TAB) أخرى في نفس عبارة (PRINT) وتأمر الحاسبة الالكترونية بالطبع في الموقع رقم (25) لنفس السطر. هذا غير جائز في لغة بيسك.

(3) استخدمت الشولة المنقوطة (;) للفصل بين دالة (TAB) والذي يأتي بعدها في عبارة (PRINT). أما إذا استخدمت الفارزة فإن الحاسبة الالكترونية ستتجاهل ما وضع بين قوسي دالة (TAB) وتطبع النتائج في مناطق طبع محددة مسبقاً.

مثال: الآن سنعود إلى برنامج قائمة أجور الهاتف لنخرج النتائج على شكل تقرير باستخدام دالة (TAB) .

```
10  REM TELEPHONE BILL
20  LET C=0
30LET T=0
40  READ N, D
50  DATA 98765, 1000
60  LET C= 6.5 + (1.75 * D)/100
70  LET T= C+ (C * 8)/100
80  PRINT TAB(5); "CUSTOMER NO." TAB(20); "COST(£)";
90  PRINT TAB(35); "TOTAL COST (£)"
100 PRINT TAB(5); "------------"; TAB(20); "---------";
110 PRINT TAB(35); "--------------"
120 PRINT TAB(5); N; TAB(20); C; TAB(35); T
130 STOP
140 END
```

نفذ البرنامج أعلاه باستخدام الحاسبة الالكترونية المتوفرة لديك. تأكد من أن النتائج ستطبع على شكل تقرير له عناوين مثلما قد صممت في عبارة (PRINT) .

اختبر نفسك :

1 ـ ما هي نتيجة كل عبارة من عبارات لغة بيسك المذكورة أدناه؟

```
60  PRINT INT(81.521+06)
50  PRINT ABS (INT ((−9.2) −0.81))
20  PRINT INT ((10 * RND)/2) +1
40  PRINT (INT ((10 * RND)/2) +1) * SGN (−6)
10  PRINT INT ((1000 * RND) +1)
```

2 ـ اكتب عبارة في لغة بيسك بحيث تولـد لك عـدداً عشوائيـاً بين العددين (1) و(100) .

3 ـ في مكتبة لغـة بيسـك هنـالـك دوال ريـاضيـة قـد بـرمجت الحـاسبـة الالكترونية لانجـازها في التعـابير الـرياضيـة بمجرد ذكـر اسمها الـذي اصـطلح عليه. اذكـر تلك الدوال الـرياضيـة مع أمثلة كعبـارات في لغة بيسك.

4 ـ مـا هي وظيفة الـدالة (TAB)؟ ومـا هي العبارة التي تستخـدمها في لغـة بيسـك؟ ثم اذكر في أي عمود (موقـع طبـع) سيبـدأ طبـع نتيجـة عبـارة (PRINT) التالية:

```
40  PRINT TAB(8.9); "BASIC"
```

5 ـ مـا هي قيمة الثـابت الرقمي الـذي سيخزن تحت اسم المتغيـر الـرقمي (A) بعد تنفيذ البرنامج التالي:

```
10  DEF FNC(X) = 6+ 4 * X ↑ 2
20  LET B=3
30  LET A= FNC(B)/10
40  PRINT "A ="; A
50  STOP
60  END
```

12. مسائل برمجة محلولة :

مسألة (1) : اكتب قيمة كل متغير رقمي نتيجة لتنفيذ كل عبارة في برنامج البيسك التالي :

X Y Z A B

```
10  READ X, Y, Z
20  LET A= X ↑ 2/ ((Y/Z) +8)
30  LET B= A * 10
40  PRINT A, B
50  STOP
60  DATA 1, 4, 2
70  END
```

الحــل : يستخدم هذا الاسلوب للتأكد من صلاحية البـرنامج بتنفيذه على بيانات اختبارية بسيطة، ومن ثم معرفة قيم المتغيرات النـاتجة عن كـل خطوة باستخدام القلم والورقة. يدعى هذا الأسلوب بالتنفيذ الجاف.

المتغيرات الرقمية					رقم العبارة
X	Y	Z	A	B	
1	4	2	—	—	10
1	4	2	0.1	—	20
1	4	2	0.1	—	30
1	4	2	0.1	1	40

1	4	2	0.1	1	50
1	4	2	0.1	1	60
1	4	2	0.1	1	70

مسألة (2) : لم يكتب برنامج البيسك أدناه بصورة كاملة حاول إكماله بحيث يستطيع قراءة أسماء ثلاثة من اصدقائك مع أرقام هواتفهم، ومن ثم يطبع الأسماء وأرقام الهاتف على شكل جدول له عناوين لتستطيع قراءته.

يرمز للأسماء بالمتغيرات الوصفية (A$) ، (B$) و(C$). ويرمز لأرقام الهاتف، على التعاقب، بالمتغيرات الرقمية (X) ، (Y) و(Z) .

```
10  READ A$ , X, ————, Y, C$, ————
20  PRINT "NAME", "PHONE NO."
30  PRINT "-------", "-----------"
40  PRINT A$ , ————
50  PRINT ————, Y
60  ————————————————
70  STOP
80  DATA "————————", ————————
90  DATA ————————————————————————
100 ————————————————————————
110 END
```

الحل : سيكون أسماء الأصدقاء مع أرقام هواتفهم كما يلي :

سامي	SAMI	٨٨٢١١٥١	8821151
زكي	ZEKI	٧٧٦٣٢١٠	7763210
أحمد	AHMAD	٤٤١٩٨٦٧	4419867

الآن يمكننا إكمال البرنامج المذكور في المسألة اعلاه.

```
10  READ A$ , X, B$ , Y, C$ , Z
20  PRINT "NAME", "PHONE NO."
30  PRINT "--------", "------------"
40  PRINT A$ , X
50  PRINT B$ , Y
60  PRINT C$ , Z
70  STOP
80  DATA "SAMI", 8821151
90  DATA "ZEKI", 7763210
100 DATA "AHMAD", 4419867
110 END
```

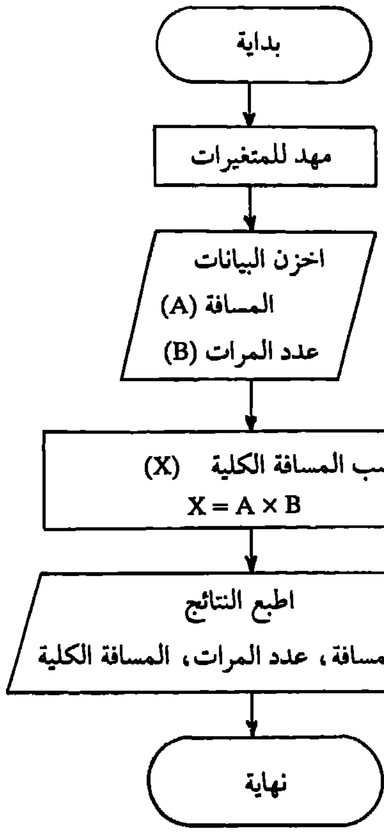
بعد تنفيذ البرنامج أعلاه ستظهر النتيجة كما يلي :

```
NAME          PHONE NO.
--------      -------------
SAMI          8821151
ZEKI          7763210
AHMAD         4419867
```

مسألة (3) : اخزن في حاسبتك الالكترونية المسافة من بيتك إلى مدرستك أو محل عملك، وعدد المرات التي ذهبت بها إلى هناك خلال سنة. احسب المسافة الكلية التي قطعتها خلال سنة. واطبع المسافة وعدد المرات التي قطعتها والمسافة الكلية. اجعل للنتائج عناوين مناسبة لتجعل نتائج البرنامج مقروءة.

الحل : لحل المسألة اعلاه باستخدام الحاسبة الالكترونية فإننا سنكتب خوارزمية الحل وستكون على شكل مخطط انسيابي. حيث تجد الخوارزمية تتكون من خطوات إدخال البيانات المتوفرة ومعالجة تلك

البيانات بعمليات حسابية ومن ثم إخراج نتائج المعـالجة تحت عنـاوين مناسبة.



الآن سننقـل خطوات خـوارزميـة الحـل إلى عبـارات بـرنـامـج بلغـة بيســك، متبعين أسلوب اختيـار العبـارة المنـاسبـة التي تؤدي الــوظيفــة المطلوبة.

(1) نمهـد للمتغيرات الـرقمية بمسـاواتها للصفـر كي نهيئها لخـزن بيـانـات البرنامج الجديد، كما يلي :

```
20 LET A=0
30 LET B=0
40 LET X=0
```

(2) يمكننا خزن (إدخال) البيانات المتوفرة

A= 3 (km) , B=200

في الحاسبة الالكترونية باستخدام ثلاثة طرق هي :

(أ) بواسطة عبارة (LET) لكون البيانات قليلة كما يلي :

```
45 LET A=3
50 LET B=200
```

(ب) بـواسـطة عبـارتي (READ + DATA) إذا كـانت البيـانـات لا تتبـدل بكثرة، كما يلي :

```
50 READ A, B
55 DATA 3, 200
```

(جـ) بـواسطة عبـارة (INPUT) إذا أردنا جعـل البرنـامج أكثـر مرونـة لتقبـل تبديل البيانات في كل مرة ينفذ بها البرنامج ، كما يلي :

```
50 INPUT A, B
```

حيث يتم إدخـال البيـانـات المتـوفـرة عنـد تنفيـذ العبـارة اعـلاه فتتغذى البيانات من قبل الجهة المستفيدة وتستمـر الحاسبـة الالكترونيـة في تنفيذ بقية عبارات البرنامج .

(3) تعالج البيانات للوصـول الى النتائج المطلوبـة بوضـع التعبير الـرياضي المناسب في عبارة (LET) ، كما يلي :

```
60 LET X= A * B
```

(4) يتم إخراج النتائج تحت عناوين مناسبة بواسطة عبارة (PRINT) فقط. ولكن يوجد أسلوبين للسيطرة على طبع النتائج.

(أ) باستخدام الفارزة أو الشولة المنقوطة للفصل بين مقاطع الرموز والمتغيرات المختلفة. يكون هذا الأسلوب سهلاً ولكن بدون مرونة في وضع النتائج بالمواقع المطلوبة.

```
70  PRINT "DISTANCE(km)", "NO. OF JOURNEYES", "TOTAL
    DISTANCE"
80  PRINT "------------", "------------", "------------"
90  PRINT A, B, X
```

(ب) باستخدام دالة (TAB) للسيطرة على مواقع الطبع مما يعطي المرونة الكافية لوضع النتائج بالشكل المناسب.

```
70  PRINT TAB(5); "DISTANCE(km)"; TAB(25);
75  PRINT "NO. OF JOURNEYES"; TAB(45); "TOTAL
    DISTANCE(km)"
80  PRINT TAB(5); "---------------"; TAB(25);
85  PRINT "---------------", TAB(45); "----------"
90  PRINT TAB(5); A; TAB(25); B; TAB(45); X
```

(5) يمكننا توضيح عبارات البرنامج بكتابة التعليق المناسب بواسطة عبارة (REM).

(6) ينهى البرنامج بعبارتي (STOP) و(END) التي يجب أن يكون لها اعلى رقم عبارة في البرنامج.

لو أردنا جعل البرنامج مرناً من حيث تقبّل بيانات مختلفة عند تنفيذه أو عند طبع النتائج فإننا نصيغ البرنامج بالعبارات المختارة التالية،

ولك الحرية في صياغته باختيـار عبارات أخـرى تكون مـوافقة لأسلوب لغـة بيسك في صياغـة العبارة والبرنامج الجيد.

```
10   REM SOLUTION OF PROBLEM NO. (3)
20   LET A=0
30   LET B=0
40   LET X=0
50   PRINT "DISTANCE (km), NO. OF JOURNEYES"
60   INPUT A, B
70   LET X= A * B
80   PRINT TAB(5); "DISTANCE(km)"; TAB(25);
90   PRINT "NO. OF JOURNEYES"; TAB(45); "TOTAL
     DISTANCE(km)"
100  PRINT TAB(5); "------------"; TAB(25);
110  PRINT "------------"; TAB(45); "----------"
120  PRINT TAB(5); A; TAB(25); B; TAB(45); X
130  STOP
140  END
```

مسـألة (4) : إذا أردت العمـل في العطلة الصيفيـة في احدى محـلات بيـع المـلابس الجاهـزة. فطلب منـك صاحب العمـل كتابـة برنـامـج بلغـة بيسـك للحاسبـة الالكترونيـة الشخصية المـوجودة في المحـل. بحيث يكون هدف هذا البرنامج هو إيجاد الكلفـة الأولية لمجمـوعة القمصـان والبنطلونات الموجودة في المحل.

إذا كان لديك المعلومات التالية:

عدد القمصان (S) = 100

سعر القميص (P1) = (5) دينار

عدد البنطلونات (T) = 150

سعر البنطلون (P2) = (10) دينار

وأن الصيغ الرياضية لحساب الكلفة الأولية هي :

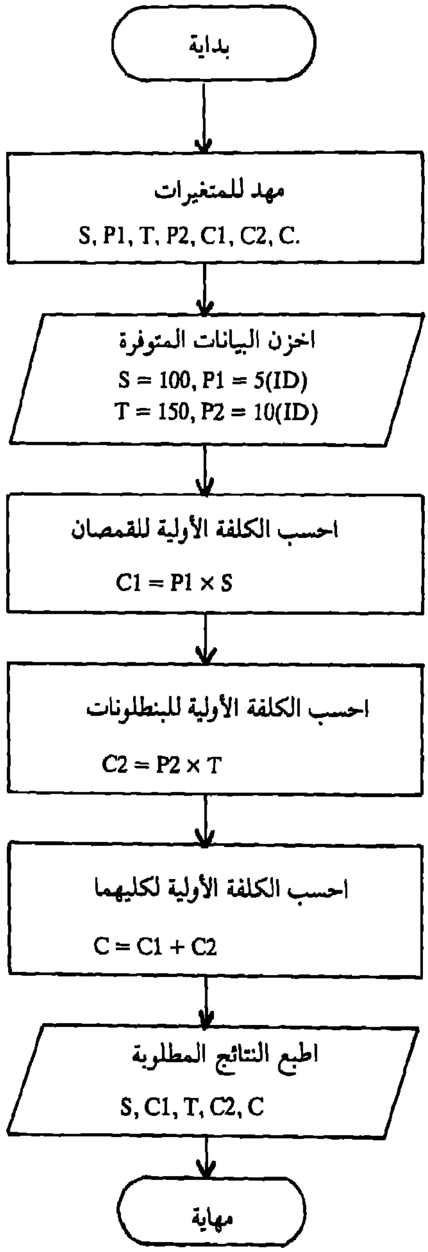
الكلفة الأولية للقمصان (C1) = P1 × S

الكلفة الأولية للبنطلونات (C2) = P2 × T

والكلفة الأولية لكليهما (C) = C2 + C1

الآن اكتب برنامجاً بحيث يمكنك تنفيذه حتى في حالة تغيير البيانات المتوفرة بدون تبديل أي من عباراته. يجب أن تكون نتائج البرنامج مسندة بالعناوين المناسبة.

الحل : يجب أولاً كتابة خوارزمية الحل ولتكن على شكل مخطط انسيابي. سيتضمن المخطط الانسيابي على أشكال هندسية لإدخال البيانات ومعالجتها ومن ثم إخراج النتائج كما يلي :


بداية
مهد للمتغيرات
S, P1, T, P2, C1, C2, C.
اخزن البيانات المتوفرة
S = 100, P1 = 5(ID)
T = 150, P2 = 10(ID)
احسب الكلفة الأولية للقمصان
C1 = P1 × S
احسب الكلفة الأولية للبنطلونات
C2 = P2 × T
احسب الكلفة الأولية لكليهما
C = C1 + C2
اطبع النتائج المطلوبة
S, C1, T, C2, C
نهاية

لجعل البرنامج ينفذ كل مرة مع بيانات مختلفة فيجب إدخال البيانات باستخدام عبارة (INPUT). تعالج البيانات باستخدام مجموعة من عبارات (LET). وتخرج النتائج تحت عناوين مناسبة باستخدام عبارة (PRINT).

```
10  REM COST REPORT
20  REM S= NO. OF SHIRTS, P1= COST/ SHIRT
30  REM T= NO. OF TROUSERS, P2= COST/ TROUSER
40  REM C1= COST OF SHIRTS
50  REM C2= COST OF TROUSERS, C= TOTAL COST
60  LET S=0
70  LET P1=0
80  LET T=0
90  LET P2=0
100 LET C1=0
110 LET C2=0
120 LET C=0
130 PRINT "NO. OF SHIRTS(S), COST/ SHIRT(P1)",
140 PRINT "NO. OF TROUSERS(T), COST/ TROUSER(P2)"
150 INPUT S, P1, T, P2
160 LET C1=P1 * S
170 LET C2=P2 * T
180 LET C=C1+C2
190 PRINT "NO. OF SHIRTS", "COST OF SHIRTS(ID)"
200 PRINT "--------------", "--------------"
210 PRINT S, C1
220 PRINT "NO. OF TROUSERS", "COST OF TROUSERS(ID)"
230 PRINT "--------------", "--------------"
240 PRINT T, C2
250 PRINT "TOTAL COST(ID)"
260 PRINT "----------------"
270 PRINT C
280 STOP
290 END
```

عند تنفيذ البرنامج اعلاه، ستظهر لك جمل لغوية لتذكرك بنوعية البيانات التي يجب طبعها. تكون تلك الجمل قبل علامة الاستفهام بسطر طبع، كما يلي :

```
NO. ȮF DHIRTS(S), COST/SHIRT(P1)
? 100, 5
NO. OF TROUSERS(T), COST/TROUSER(P2)
? 150, 10
```

بعد ذلك تستمر الحاسبة الالكترونية في تنفيذ بقية عبارات البرنامج اعلاه حتى تظهر النتائج معنونة بالشكل التالي :

```
NO. OF SHIRTS                    COST OF SHIRTS (ID)
------------------------------   ------------------------------
100                              500

NO. OF TROUSERS                  COST OF TROUSERS (ID)
------------------------------   ------------------------------
150                              1500

TOTAL COST (ID)
------------------------------
2000
```

مسألة (5) : أوجد نتيجة البرامج البسيطة التالية :

(أ)

```
10  DEF FNA (X) = X/2
20  READ V, W, Y
30  LET S= INT (FNA(V))
40  LET T= INT (ABS (FNA (W)
50  LET R= FNA(Y)
60  PRINT S, T, R
70  STOP
80  DATA 7, −9, 10
90  END
```

الحـل : سنتبـع القيم الرقميـة المخزونـة تحت اسم كل متغيـر من خـلال تنفيذ كل عبارة حتى نصل الى النتيجة النهائية .

V	W	Y	S	T	R	رقم العبارة
						المتغيرات الرقمية
—	—	—	—	—	—	10
7	9	10	—	—	—	20
7	9	10	3	—	—	30
7	9	10	3	4	—	40
7	9	10	3	4	5	50
7	9	10	3	4	5	60

(ب)

```
10  READ A
20  LET X= SGN (A) * INT (ABS (Q)/3)
30  PRINT "A ="; A, "X ="; X
40  STOP
50  DATA −987. 543
60  END
```

الحل : ستكون نتيجة البرنامج اعلاه كما يلي :

A =− 987.543   X = − 329

يمكنك التأكد من النتيجة اعلاه بالتنفيذ الجاف للبـرنامـج أولاً ، ومن ثم تنفيذه على الحاسبة الالكترونية .

(جـ)

```
10  LET A= 1000
20  LET B = (INT ((A * RND+1)/4)
30  PRINT "B="; B
```

```
40 STOP
50 END
```

الحـــل : في كل مرة ينفذ بها البرنامـج اعلاه سينتـج عدداً يقـع بين الصفر و(250) . وذلك نتيجة لوجود الدالة (RND) في التعبيـر الريـاضي الذي تعالج بواسطته البيانات.

(د)

```
10 DEF FND(Z) = (SQR (Z+3( * EXP(Z))
20 LET X=1
30 LET Y=FND(X)
40 PRINT "Y="; Y
50 STOP
60 END
```

الحل : ستظهر نتيجة البرنامج اعلاه كما يلي :

```
T = 5.42656
```

(هـ)

```
10 DEF FNA(X) = ABS (SIN (−P/2)/TAN (P/4)
20 LET P = 3.14159
30 LET X = 1000
40 LET Y = FNA(X)
50 PRINT "Y="; Y
60 STOP
70 END
```

الحل : لاحظ أن الحجة في الدالة المعـرفة (FNA) هي ليست ذات أهميـة في هـذا البرنـامج . لـذا ستجـد النتيجـة لا تعتمـد على قيمـة (X) التي اتخذتها في البرنامج اعلاه. ستظهر نتيجة البرنامج اعلاه:

```
Y = 1
```

13. خلاصة :

يعتبر هذا الفصل دعوة للقارىء لبدء البرمجة بلغة بيسك. فقد قدمنا لك عشرة عبارات أساسية من لغة بيسك، يمكنك بواسطتها برمجة حل طيف واسع من المسائل الاكاديمية والتجارية.

إن الفكرة الرئيسية من وراء طرح موضوعات هذا الفصل هو تعليم القارىء كيفية نقل خطوات خوارزمية حل إلى عبارات برنامج بلغة بيسك، ليستطيع تنفيذ الحل بواسطة أداة العصر (الحاسبة الالكترونية). تلخص النقاط التالية أهم أفكار هذا الفصل؛

أولاً : تستخدم عبارة (LET) لتخصيص قيم للمتغيرات عن طريق إما الادخال المباشر أو كنتيجة لتنفيذ العلمات في تعبير حسابي. ومن الملاحظات المهمة حول عبارة (LET) :

(أ) تخصص عبارة (LET) قيمة رقمية لمتغير رقمي أو ثابت مقطعي لمتغير مقطعي. تكون البيانات في الجهة اليمنى من علامة المساواة والمتغيرات في الجهة اليسرى.

(ب) يمكن أن تكون المتغيرات في عبارة (LET) ذات أبعاد.

(جـ) لا تخزن علامة قوسي الاقتباس المحيطة بالثابت المقطعي في ذاكرة الحاسبة الالكترونية كجزء من بيانات المتغير المقطعي.

(د) في معظم لهجات لغة بيسك يمكنك حذف الكلمة الأساسية (LET) من العبارة دون الاخلال بوظيفة العبارة لتخصيص البيانات أو تنفيذ العمليات الرياضية في تعبير رياضي.

ثانياً : تستخدم عبارة (PRINT) لاظهار النتائج في لغة بيسك، وهي تتألف من رقم العبارة والكلمة الأساسية (PRINT) ومعلمة أو أكثر. من الملاحظات المهمة حول عبارة (PRINT) :

(أ) يتم الفصل بين المتغيرات في عبارة (PRINT) باستخدام الفارزة أو الشولة المنقوطة. تعمل الفارزة على نقل طبع النتائج الى منطقة طبع جديدة. أما الشولة المنقوطة فتؤيد على استمرار الطبع بدون الانتقال الى منطقة طبع جديدة.

(ب) يمكنك استخدام الدالة (TAB) في عبارة (PRINT) للسيطرة على طبع النتائج في المواقع التي يريدها المبرمج بدون تقيده بمناطق الطبع.

(جـ) يتم إحاطة الثابت الوصفي بقوسي الاقتباس في عبارة (PRINT) ولكن عند طبع النتائج سيظهر الثابت الوصفي بدون قوسي الاقتباس.

(د) يمكن استخدام عبارة (PRINT) لايجاد قيمة تعبير حسابي حيث تطبع النتيجة الرقمية بالمواصفات التالية:

(1) إذا كان العدد صحيحاً لا تطبع أصفار بعد العلامة العشرية.

(2) هنالك حدود دقة لا يمكن تجاوزها.

(3) تمثل الأعداد الصغيرة جداً والكبيرة جداً باستخدام الشكل الأسي.

(4) يترك فراغ بدل علامة الزائد في العدد الموجب.

(5) يترك فراغ بعد إكمال طبع العدد إذا كانت الشولة المنقوطة هي الفاصلة عن الحد التالي في عبارة (PRINT) .

(هـ) يتم ترجمة رمز متوازي الأضلاع عند إخراج النتائج في المخطط الانسيابي الى عبارة (PRINT) في لغة بيسك.

ثالثاً : تكون آخر عبارة في برنامج البيسك هي عبارة (END) .

رابعاً : بواسطة عبارة (REM) يتم توضيح الهدف من البرنامج وتعريف المتغيرات المستخدمة فيه وتجهيز فواصل مرئية بين العبارات.

خامساً : يتم تخصيص البيانات المحتواة في عبارة (DATA) الى متغيرات موجودة في عبارة (READ). وبهذه الطريقة يتم تغذية البيانات لتخزن في الحاسبة الالكترونية. من الملاحظات المهمة حول عبارتي (READ + DATA) :

(أ) يمكن ذكر البيانات في أكثر من عبارة (DATA) وفي مواقع مختلفة من البرنامج لنفس عبارة (READ). والعكس صحيحاً أيضاً.

(ب) يجب أن تتطابق نوعية البيانات في عبارة (DATA) مع نوعية المتغيرات في عبارة (READ) .

(1) يتم إهمال البيانات الفائضة عن عدد المتغيرات.

(2) تظهر رسالة خطأ إذا لم يكن هنالك بيانات كافية في عبارة (DATA) تغطي المتغيرات في عبارة (READ) .

(جـ) تعيد عبارة (RESTORE) الحاسبة الالكترونية الى بداية صف البيانات الموجودة في عبارات (DATA). بحيث عندما تنفذ الحاسبة الالكترونية عبارة (READ) التالية لعبارة (RESTORE) فإنها تخصص قيم للمتغيرات من بداية صف البيانات وتترك الباقي منها لعبارات (READ) لاحقة.

سادساً : تسمح عبارة (INPUT) للمستفيد أو المبرمج بادخال البيانات أثناء تنفيذ البرنامج، ولهذا تطبيقات مثيرة منها في الألعاب ومحاكات بعض الفعاليات مثل تعليم قيادة السيارة أو اختبار الطلاب باستخدام الحاسبة الالكترونية. يمكن ذكر بعض الملاحظات المهمة حول عبارة (INPUT) :

(أ) **تعمل عبارة** (INPUT) مثل عبارة (READ) ولكن بدون وجود عبارة (DATA) مرافقة، حيث يتم إدخال البيانات أثناء تنفيذ البرنامج، وبعد ظهور إشارة متفق عليها على جهاز إدخال البيانات.

(ب) يكون طلب إدخال البيانات إما بواسطة صوت موسيقي ينبه المستفيد، أو ظهور علامة استفهام (?) أو سهم (←) أو رمز أكبر من (>) .

(جـ) يمكن أن تطلب عبارة (INPUT) تخصيص بيانات لأكثر من متغير. تفصل الفارزة بين المتغيرات.

(د) يمكن إظهار جمل توضيحية قبل إشارة إدخال البيانات بوضعها في عبارة (PRINT) يكون موقعها قبل عبارة (INPUT) في برنامج بيسك.

(هـ) يجب مطابقة نوع البيانات المدخلة مع نوع المتغيرات في عبارة (INPUT).

سابعاً : في المخطط الانسيابي، يكون رمز إدخال البيانات على شكل متوازي أضلاع. وفي لغة بيسك، تغذى البيانات المتوفرة في رمز متوازي الأضلاع الى الحاسبة الالكترونية بواسطة ثلاثة طرق، تكون لكل منها شروط وخصائص:

(1) بواسطة عبارة (LET) ، إذا كانت البيانات قليلة العدد.
(2) بواسطة عبارتي (READ + DATA)، إذا كانت البيانات كثيرة وغير قابلة للتغيير بكثرة.
(3) بواسطة عبارة (INPUT) ، إذا أردنا تنفيذ البرنامج على بيانات تتغير باستمرار، وبدون تغيير أي من عبارات ذلك البرنامج.

ثامناً : تستطيع لغة بيسك إيجاد قيم أحد عشر دالة شائعة الاستعمال في الرياضيات، وبمجرد ذكر اسمها الذي اصطلح عليه. تدعى تلك الدوال بدوال مكتبة لغة بيسك. تقسم تلك الدوال الى:

(1) الدوال المثلثية: SIN(X), COS(X), TAN(X), ATN(X)
(2) الدوال الأسية: SQR(X), EXP(X), LOG(X)
(3) الدوال الرياضية: SGN(X), ABS(X), INT(X), RND

تاسعاً : بإمكان المبرمج استخدام عبارة (DEF) لتعريف الدوال الرياضية التي لا تتوافر في مكتبة لغة بيسك. يمكن الرجوع الى الدالة الرياضية المعرّفة بذكر اسمها فقط. واسم الدالة يتكوُن من الرمزين (FN) متبوع بحرف مناسب من حروف هجاء اللغة الانكليزية، ثم يتبع بقوسين يكتب بداخلهما حجة للدالة الرياضية لكي تكون متغيرها المستقل.

14. كلمات قد تعلمتها :

مفردة من لغة	Vocabulary
قواعد نحو	Syntax
عبارة (اجعل)	LET
عدّاد	counter
انشوطة	loop
عبارة (اطبع)	PRINT
عنونة	Heading
عبارة (توقف)	STOP
عبارة (النهاية)	END
عبارة (الملاحظة)	REM
عبارة (بيانات)	DATA
عبارة (اقرأ)	READ
عبارة (أعد خزن)	RESTORE
عبارة (ادخل)	INPUT
محاكاة	Simulation
دوال مكتبة لغة بيسك	Library Functions
حجة، بينة دالة	Argument
دوال مثلثية	Trigonometric Functions
دوال أسية	Exponential Functions
دوال رياضية	Mathematical Functions

Radian	زاوية شعاعية
Degree	درجة
DEF	عبارة (عرّف)
Tabulating Function	دالة الجدولة

15. تمارين (3) :

استخدم كلمات قد تعلمتها .

(أ) الحجة	(و) عبارة (LET)
(ب) الزاوية الشعاعية	(ز) عبارة (INPUT)
(جـ) الدالة (RND)	(ح) عبارة (LET)
(د) دالة (TAB)	(ط) عبارة (END)
(هـ) عبارة (READ)	(ى) دالة (ABS)

1 ـ تستخدم ـــــــــ لطبع النتائج على مواقع طبع مخصصة من سطر الطبع .

2 ـ تسمح ـــــــــ بإدخال البيانات أثناء تنفيذ البرنامج .

3 ـ تستخدم ـــــــــ مع الدوال المثلثية في الحاسبة الالكترونية .

4 ـ تدخل البيانات في برنامج بيسك بواسطة ـــــــــ إذا كانت قليلة العدد .

5 ـ تختلف تركيبة واستجابة ـــــــــ من حاسبة الكترونية الى أخرى .

6 ـ ـــــــــ هي القيمة التي يجب إيجادها بواسطة دالة رياضية .

7 ـ تدخل البيانات في برنامج بيسك بواسطة ـــــــ إذا كانت البيانات كثيرة جداً.

8 ـ تسبب ـــــــ توقف تنفيذ البرنامج في الحاسبة الالكترونية.

9 ـ تهمل ـــــــ إشارة العدد وتأخذه هو فقط بنظر الاعتبار.

10 ـ تستخدم ـــــــ لتخصيص القيم الى المتغيرات.

صح أم خطأ :

1 ـ تنفذ عبارة (REM) من قبل الحاسبة الالكترونية لتخرج الملاحظات مع نتائج البرنامج.

2 ـ تتضمن دوال مكتبة لغة بيسك كل من

SIN(X), LOG(X), FND(X)

3 ـ تولّد دالة (RND) عدداً بصورة عشوائية بين الصفر (0.000000) والواحد (1.000000).

4 ـ تعرّف الدالة التي يحتاجها المبرمج وليس احدى دوال مكتبة لغة بيسك بواسطة عبارة (DEF).

5 ـ يبدأ اسم الدالة التي تعرّف بعبارة (DEF) بالرمزين (FN).

6 ـ تستخدم عبارة (READ) لإدخال البيانات التي تتغير بكثرة.

7 ـ يمكن ترك سطر طبع فارغاً في نتائج برنامج بيسك بواسطة عبارة (PRINT) بدون معلمة.

8 ـ تسبب الشولة المنقوطة طبع النتائج في مناطق طبع محددة مسبقاً.

9 ـ يمكن استخدام الفارزة مع دالة (TAB) .

10 ـ لا يمكن استخدام الفارزة والشولة المنقوطة في عبارة (PRINT) واحدة.

أجب باختصار :

1 ـ هنالك ثلاثة طرق لادخال البيانات في لغة بيسك. اذكرها، وتحت أية شروط يمكن استخدام كل منها.

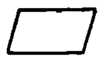
2 ـ تعتبر العبارات التالية غير صحيحة من وجهة نظر قواعد نحو لغة بيسك. صححها.

```
90  LET 15=F
10  PRINT TAB; (16); X
20  LET TAB (45); H$
30  READ TAB(10); A
70  DATA 10, Z$ , 100
```

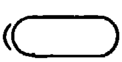
3 ـ ما الفرق بين الدالة (SGN) والدالة (ABS) ؟

4 ـ كيف تترجم رموز المخطط الانسيابي التالية الى عبارات في لغة بيسك؟

(أ) رمز معالجة البيانات ()

(ب) رمز إدخال البيانات ()

(جـ) رمز إخراج النتائج ()

(د) الرمز الطرفي لانهاء البرنامج ()

5 ـ اكتب عبارة (READ) واحدة لتخصيص البيانات الموجودة في عبارات (DATA) التالية الى متغيرات مناسبة :

```
10  DATA 10, "BASIC", 20
20  DATA "AUGUST", 30, "SUNDAY"
30  DATA 99.9 e−19, 56.7, 8.11
```

6 ـ يمكنك كتابة ملاحظة أو تعليق بواسطة عبارة (REM) وعبارة (PRINT). اذكر خصائص استخدام كل منها.

7 ـ اكتب عبارة في لغة بيسك بحيث تختار لك رقم الحظ من بين أرقام المدى (0 - 9) وبدون تحيز.

8 ـ ما هي نتيجة العبارة التالية :

```
20  PRINT "X="; INT ((10 * RND)/5) +1
```

9 ـ في أي موقع طبع سيبدأ طبع الكلمة (GOOD) في عبارة (PRINT) التالية :

```
30  PRINT TAB(9.9); X; TAB(40.1); "GOOD"
```

10 ـ ما هي قيمة الثابت الرقمي الذي سيخزن تحت اسم المتغير الرقمي (A) بعد تنفيذ البرنامج التالي :

```
10  DEF FND(X) =INT (SQR(X)) + ABS )−9.2)
20  LET B=0
30  LET Z= FND(B) ↑ 5
40  PRINT "Z="; Z
50  STOP
60  END
```

اختار الجواب الصحيح :

1 ـ ما هي نتيجة البرنامج التالي :

```
10  LET A=4
20  LET B=A+6
30  PRINT "B="; A+B/2 -1
40  STOP
50  END
```

(أ) B=16

(ب) B=20

(جـ) B=8

(د) B=5.53

2 ـ تنفيذ أي من أجزاء البرامج التالية سينتج عنه رسالة خطأ؟ اذكر السبب.

(أ)

```
10  READ X, Y
20  DATA 9, 8, 7
```

(ب)

```
10  READ X, Y
20  READ Z
30  DATA 9, 8, 7
```

(جـ)

```
10  READ X, Y, Z
20  DATA 9, 8
30  DATA 7
```

(د)

```
10  DATA 9
20  DATA 7
30  READ X, Y
```

3 ـ ماذا سيخزن في المتغيرات (A) ، (B) ، (D$) و(C) بعد تنفيذ جزء البرنامج التالي :

```
10  READ A, B, D$, C
20  READ B
30  READ C, D$
40  DATA 1, 2
50  DATA "GOOD"
60  DATA 3, 4
70  DATA 5, "BASIC"
```

	A	B	D$	C
(أ)	1	2	GOOD	3
(ب)	3	2	BASIC	4
(جـ)	5	3	GOOD	1
(د)	1	4	BASIC	5

4 ـ ما الذي سيطبعه البرنامج التالي :

```
10  READ A, B
20  LET X= A/B
30  READ A, B
40  LET X= A * B
50  PRINT "X="
60  STOP
70  DATA 7, 2, 1, 10
80  END
```

(أ) 271

(ب) X=

(جـ) 1102

(د) 1

5 ـ ما هي نتيجة البرنامج التالي :

```
10  READ A, B$
20  READ C, D
```

```
30  PRINT C, D
40  READ A, B$, C, D
50  PRINT C/D, B$
60  STOP
70  DATA 9, "BASIC", 8, 7, 6, "GOOD", 4, 2
80  END
```

(أ) 7 8
GOOD 2

(ب) 8 9
BASIC 7

(جـ) 4 6
GOOD 2

(د) 4 2
BASIC 6

6 ـ ماذا سيطبع البرنامج التالي :

```
10  READ X, Y
20  LET X= X+Y
30  LET Y= Y+10
40  LET Z= X+Y
50  PRINT Z
60  DATA 1, 2, 3, 4
70  STOP
80  END
```

(أ) 3
(ب) 12
(جـ) 15
(د) 4

7 ـ ما هي نتيجة البرنامج التالي:

```
10  LET V=1
20  LET V=V+10/2
30  PRINT V
40  STOP
50  END
```

(أ) 1

(ب) 2

(جـ) 10

(د) 6

8 ـ في عبارة (PRINT)، يمكنك الطبع في مناطق طبع مخصصة مسبقاً باستخدام:

(أ) دالة (TAB) .

(ب) الفارزة.

(جـ) الشولة المنقوطة.

(د) علامة الاستفهام.

9 ـ إن وجود الشولة المنقوطة في نهاية عبارة (PRINT) سيسبب:

(أ) طبع نتائج عبارة (PRINT) في سطرين الواحد بعد الآخر.

(ب) طبع نتائج عبارة (PRINT) التالية على نفس سطر الطبع.

(جـ) ظهور رسالة خطأ.

(د) ترك سطر طبع فارغ بدون أن يطبع فيه شيء.

10 ـ تتكون مكتبة لغة بيسك من:

(أ) احد عشر دالة رياضية.

(ب) مائة دالة رياضية .

(جـ) عشرة دوال رياضية .

(د) دالة رياضية واحدة هي FND(X) .

الفصل الرابع

كيفية التعامل مع الحاسبة الالكترونية لتنفيذ برنامج بيسيك

1. مقدمــة
2. ابتداء الاتصال بالحاسبة الالكترونية
3. خلق وتنفيذ برنامج بيسك
4. خزن برنـامج بيسك

اختبر نفسك

5. إعادة طبع برنامج بيسك
6. تنقيح برنامج بيسك

اختبر نفسك

7. مسائل محلولة
8. خـلاصـــة
9. كلمات قد تعلمتها
10. تمـارين (4)

1. مقدمة :

بعد أن تعلمت كيفية كتابة الخوارزمية (الفصل الأول) وكيفية كتابة برنامج بسيط في لغة بيسك (الفصل الثاني والثالث). فمن الأفضل الآن أن نتعلم كيفية تنفيذ برنامج بيسك على حاسبة الكترونية، والتي تعتبر المرحلة الثالثة والأخيرة في مهمة حل مسألة أكاديمية بواسطة الحاسبة الالكترونية.

يشتمل تنفيذ برنامج البيسك على الاتصال بالحاسبة الالكترونية أولاً، خزن البرنامج واستدعائه عند الحاجة، إعادة طبعه، ومن ثم تنقيحه إذا وجدت هناك أخطاء أو إضافات أو حذف. نستطيع إنجاز جميع المهمات المذكورة اعلاه بواسطة مجموعة أوامر (COMMANDS) متعارف عليها في لغة بيسك. فما عليك سوى تعلم وظيفة كل أمر لتتمكن من تنفيذ برنامج بيسك بسهولة ومهارة، بحيث تصبح العملية ممتعة أكثر منها مشقة.

سيتعلم القارىء من دراسته لهذا الفصل الأمور المهمة التالية لتنفيذ برنامج بيسك على الحاسبة الالكترونية:

1 ـ الأوامر (COMMANDS) في لغة بيسك.

2 ـ بداية الاتصال بالحاسبة الالكترونية، وإنهاء ذلك الاتصال.

3 ـ إيجاد برنامج بيسك.

4 ـ خزن برنامج بيسك في الذاكرة الاحتياطية للحاسبة الالكترونية.

5 ـ تنفيذ برنامج بيسك على حاسبة الكترونية.

6 ـ تنقيح برنامج بيسك.

2. ابتداء الاتصال بالحاسبة الالكترونية :

لكل حاسبة الكترونية قواعدها الخاصة بها لابتداء (SIGN ON) أو إنهاء (SIGN OFF) الاتصال بها من قبل الجهة المستفيدة. لذا ارجع إلى كراسة تعليمات حاسبتك الالكترونية والخاصة بالمبرمج، لتعرف تلك القواعد. فإذا كنت تشتغل على حاسبة الكترونية شخصية، فبمجرد توصيلها بالقوة الكهربائية سيكون بإمكانك إدخال برنامج البيسك لأن نظام لغة بيسك قد خزن مباشرة كجزء من البرامجيات المتواجدة في ذاكرة الحاسبة الالكترونية الشخصية.

أما إذا كنت تشتغل على حاسبة الكترونية متوسطة أو كبيرة الحجم، فسيكون اتصالك المباشر بجهاز طرفي (Terminal) . وستجد قواعد خاصة لابتداء أو إنهاء الاتصال من حاسبة الكترونية الى أخرى. وبالرغم من وجود تلك الاختلافات، يمكننا وضع الاجراءات العامة والمشتركة بين تلك الحاسبات:

1 ـ شغّل الجهاز الطرفي، (أو اطلب رقم مكتب الحاسبة الالكترونية إذا كان الجهاز الطرفي متصل بواسطة خط هاتف).

2 ـ سيظهر لك على شاشة أو جهاز طبع المحطة الطرفية رسالة من نظام

التشغيل (operating System) ، تسألك طبع بعض المعلومات التعريفية ليتأكد نظام التشغيل من أنك جهة مخولة للاتصال بها .

3 ـ كما تسألك لطبع رقم حسابك (account number) المصرفي لتستقطع منه أجور استخدامك لوقت الحاسبة الالكترونية .

4 ـ تدخل برنامج البيسك إلى الحاسبة الالكترونية وتستحصل نتائج ذلك البرنامج .

5 ـ تستطيع إنهاء الاتصال بالحاسبة الالكترونية باستخدام الأمر (BYE) .

في الحاسبات الالكترونية الكبيرة، يشترك أكثر من جهة مستفيدة لتنفيذ برامجها في نفس الوقت. تدعى هذه الطريقة بنظام المشاركة الزمنية (Time Sharing System) حيث تجد الحاسبة الالكترونية تتحاور مع الجهة المستفيدة كما يلي :

```
YOU ARE CONNECTED TO U.O.T COMPUTER SYSTEM.
PLEASE TYPE YOUR ACCOUNT NUMBER
? 91827
PLEASE TYPE YOUR PASSWORD
? SAMI 1986
```

تطبع معظم الحاسبات الالكترونية فوق كلمة السر (PASSWORD) لتحاشي معرفتها من قبل الفضوليين أو إمكانية استخدامها من قبل جهة غير مخولة .

```
PLEASE TYPE YOUR PASSWORD
 ?
 USER 91827 SIGNED ON AT 10.30 HRS
 WHAT LANGUAGE
 ? BASIC
```

بعـد إنهاء عمليـة التعامـل مع الحـاسبة الالكتـرونية لتنفيـذ بـرنـامـج البيسك، وعندما تريد إنهاء اتصالك مـع الحاسبـة الالكترونيـة، فما عليـك سـوى طبع الأمـر (BYE) ، عندهـا تظهـر لـك رسـالـة تبين فيهـا الحـاسبـة الالكترونية وقت إنهـاء الاتصال والـوقت المستغرق في الاتصـال بين الجهة المستفيـدة والحاسبـة الالكترونيـة، وفي بعض أنواع الأنـظمة يـظهـر أيضـاً المبلغ الذي سيقتطع من الجهة المستفيدة.

```
BYE
USER 91827 SIGNED OFF AT 10.40 HRS
TIME CONNECTED = 10 MINS
```

اعلم أن مـا ذكر أعـلاه يمثل الاجـراءات المشتركـة بين أنـظمـة لغـة بيسـك لابتداء وإنهـاء الاتصال بـالحاسبـة الالكترونيـة لمعـالجـة بـرنـامـج بيسـك. فادرس تلك الاجـراءات في كـراسـة تعليمـات المبـرمـج التـابعـة لحاسبتك الالكترونية.

3. خلق وتنفيذ برنامج بيسك جديد :

تكون الحاسبة الالكترونية جاهزة لاستلام برنامج بيسك جـديد بعـد إتمام اجراءات ابتداء الاتصال بهـا. يستلم برنـامج البيسـك بطبـع عباراتـه الـواحدة بعـد الأخرى، حيث تبـدأ كـل منهـا برقم العبـارة، وينتهي إدخال العبـارة بالضغط على مفتـاح (RETURN). أنظر لـوحـة مفـاتيـح حـاسبتـك الشخصية أو الجهاز الطرفي الذي تدخل بواسطته برنامجك.

تجبر إجراءات الاتصال بالحاسبة الالكترونية على أن يكـون أول أمر (Command) لخلق برنامج بيسك هو الأمر (NEW) أي (بـرنامـج جديـد). لاحظ التحـاور التـالي بين نـظام التشغيـل لحـاسبـة الكتـرونيـة والجهـة المستفيدة (كتبت إجابتها باللون الأحمر) :

```
WHAT LANGUAGE
? BASIC
NEW PROGRAM OR OLD
? NEW
GIVE NEW PROGRAM A NAME
? PROGM 1
BASIC IS READY
```

عند ظهور الجملة BASIC IS READY تكون الحاسبة الالكترونية جاهزة لخزن عبارات البرنامج الجديد (PROGM1) بتهيئة جزء من الذاكرة الرئيسية فيها.

```
BASIC IS READY
10  REM PROGM 1
20  READ A, B, C
30  LET F=(A+B)/C
40  PRINT "F="; F
50  STOP
60  DATA 9, 7, 8
70  END
```

الآن بعد اتمام استلام البرنامج بقي علينا تنفيذه بطبع الأمر (RUN) أي (نفّذ).

```
RUN
F=2

BASIC IS READY
```

ظهرت نتيجة البرنامج (PROGM1) مباشرة نتيجة لعدم ارتكاب أخطاء عند إدخال البرنامج. تستمر الحاسبة الالكترونية في تنفيذ البرنامج إلى أن يحدث أحد الأمور التالية:

1 ـ اتفيذ عبارة (STOP) أو عبارة (END) .

2 ـ خطأ في قواعد نحو (syntax error) عبارة في برنامج بيسك.

3 ـ خطأ ناتج عند تنفيذ (execution error) تعبير رياضي، كالقسمة على صفر.

4 ـ خطأ في أجهزة الحاسبة الالكترونية (hardware error) .

4. خزن برنامج بيسك :

عرفت أن الأمر (NEW) يهيء جزء من الذاكرة الرئيسية للحاسبة الالكترونية لاستقبال برنامج بيسك جديد. يمكن الاحتفاظ بنسخة من برنامج البيسك في الذاكرة الاحتياطية لكي نتمكن من استدعائها في أوقات أخرى بدلاً من طبع البرنامج في كل مرة نريد تنفيذه فيها. والأمر المستخدم لذلك هو (SAVE) أي (اخزن). لنخزن البرنامج (PROGM1) .

```
SAVE PROGM 1
```

الآن تم نقل نسخة من برنامج (PROGM 1) من الذاكرة الرئيسية إلى الذاكرة الاحتياطية. وبذلك عند إنهاء الاتصال بالحاسبة الالكترونية سوف لا يمحى هذا البرنامج بل يمكن استدعائه عند ابتداء الاتصال مرة أخرى. ولكن كن حذراً إذا كنت ممن ينفذ برامجه بالاتصال مع مكتب حاسبة الكترونية كبيرة لأن خزن البرنامج في الذاكرة الاحتياطية سيكلفك دفع أجراً. يستقطع عن كل يوم يبقى به البرنامج مخزوناً. لذا يمكنك حذف البرامج التي لا تستحق دفع أجر من أجل خزنها باستخدام الأمر (UNSAVE) أي (امحو). فلو كان لديك ثلاثة برامج تريد حذفهم من

الـذاكرة الاحتيـاطية للحـاسبة الالكتـرونيـة (PROGM 2) ، (PROGM 3) ، (PROGM 4) فيمكنك استخدام الأمر التالي :

```
UNSAVE PROGM2, PROGM3, PROGM4
```

اختبر نفسك :

1 ـ هل إجراءات ابتـداء الاتصال وإنهـائه بـالحاسبـة الالكترونيـة الشخصية هي نفسها مع حاسبة الكترونية كبيرة الحجم؟ اذكر الفرق.

2 - ما هو الأمر الذي ينهي الاتصال بالحاسبة الالكترونية؟

3 ـ تعمـل الحاسبـة الالكترونيـة على إخفاء كلمـة السـر التي تـطبعهـا أنت كجهة مخولة بالعمل معها. اذكـر طريقـة واحدة مشهـورة لاخفاء كلمـة السر.

4 ـ مـا هو المفتـاح الذي يجب ضغـطه عند إدخـال سطر جـديد الى ذاكـرة الحاسبة الالكترونية؟ ولماذا يجب أن يضغط؟

5 ـ اذكر الأمر (COMMAND) الذي يستخدم لأداء كل وظيفة مما يلي:

(أ) خزن برنامج بيسك في الذاكرة الاحتياطية .

(ب) خلق برنامج بيسك جديد .

(جـ) تنفيذ برنامج بيسك .

(د) محو برنامج بيسك من الذاكرة الاحتياطية .

5. إعادة طبع برنامج بيسك :

إذا أردت في وقت لاحق ابتداء الاتصال بالحاسبـة الالكترونيـة لتنفيذ البرنامج (PROGM 1) مرة أخرى. فعليك أولاً استـدعاء ذلـك البرنـامج من الذاكرة الاحتيـاطية، ومن ثم إعـادة طبعه لإجـراء اللازم من تنقيحـات على ذلك البرنامج. يستخدم الأمر (OLD) لاستدعاء البرنامج:

```
OLD  PROGM 1
```

ويستخدم الأمر (LIST) أي (أعـد طبع) لإظهـار البرنـامج مـرة أخرى على المحـطة الطرفيـة أو الحاسبـة الالكترونيـة الشخصية. لنعـود الآن الى الخطوة الأخيرة من ابتـداء الاتصال حيث تجـد نظام التشغيـل يسأل الجهـة المستفيدة عن البرنـامج الـذي تريـد تنفيذه هـل هو جـديد ليخلق أم قـديم فيجلب:

```
NEW PROGRAM OR OLD
? OLD PROGRAM 1
BASIC IS READY
LIST
10   REM PROGM 1
20   READ A, B, C
30   LET F= (A + B)/C
40   PRINT "F="; F
50   STOP
```

```
60  DATA 9, 7, 8
70  END
```

لاحظ أن لإعادة طبع برنامج قديم فوائد منها متابعة الأخطاء الموجودة في البرنامج ومشاهدته كيف تم خزنه في الذاكرة الاحتياطية للحاسبة الالكترونية.

هنالك أمر (CATALOG) خاص بإعادة طبع ما خزن من برامج تحت رقم الحساب الذي تعرف به جهة مستفيدة معينة. فلو كان لديك أربعة برامج مخزونة في الذاكرة الاحتياطية للحاسبة الالكترونية، وأردت معرفة أسمائها فما عليك سوى طبع الأمر (CATALOG) وسيظهر كما يلي:

```
CATALOG
USER 91827 PROGRAMS STORED

PROGM 30
PROGM 40
PROGM 50
PROGM 60
```

6. تنقيح برنامج بيسك :

أسباب كثيرة تؤدي الى حدوث أخطاء عند طباعة برنامج باستخدام لوحة مفاتيح حاسبة الكترونية. فلا تقلق عند حدوث مثل هذا الخطأ فإن تصحيحه بسيط، بحيث يتم تغيير أو تنقيح تلك العبارة. عند التنقيح، يمكن أن تقوم بأحد الأمور التالية :

(1) تصحيح أو استبدال عبارات موجودة في البرنامج .

(2) حذف عبارات .

(3) إقحام عبارة جديدة بين عبارتين .

يتم تصحيح أو استبدال عبارة في برنامج بيسك بطبع رقم العبارة وكتابة العبارة الصحيحة أو البديلة ومن ثم الضغط على مفتاح (RETURN) الموجود ضمن مفاتيح لوحة مفاتيح الحاسبة الالكترونية .

أما لحذف عبارة في برنامج بيسك فهنالك طريقتين :

(أ) نطبع رقم العبارة فقط ومن ثم نضغط على المفتاح (RETURN) .

(ب) نستخدم الأمر (DELETE) أي (احذف) ثم نطبع رقم العبارة التي يراد حذفها بعدها نضغط على المفتاح (RETURN) .

```
DELETE 140
```

سيسبب الأمر اعلاه حذف العبارة رقم (140) من برنامج البيسك.

لإقحام عبارة جديدة، نعطي تلك العبارة رقماً يقع تسلسله بين رقمي عبارتين محددتين، ثم نطبع العبارة ونضغط على المفتاح (RETURN). ولاعادة ترقيم عبارات البرنامج جميعها بعد عملية إقحام العبارة الجديدة، نستخدم الأمر (RENUMBER) ، أو (RESEQUENCE) .

لنفترض أننا عند إدخال البرنامج (PROGM 1) قد ارتكبنا أخطاء بحيث يتوجب علينا تنقيحه لكي يعطي النتائج المطلوبة، لذا سنجري عليه التصحيحات المطلوبة باستخدام الوسائل التي زودتنا بها أوامر لغة بيسك، وكما يلي :

```
BASIC IS READY
10  AEM PROGM 1
20  READ A, B, C
30  PRINT "F="; F
40  END
50  STOP
60  DATA
70  END

DELETE 40
BASIC IS READY
10  REM PROGM 1
25  LET F= (A+B)/C
60  DATA 9, 7, 8

LIST
10  REM PROGM 1
20  READ A, B, C
25  LET F= (A + B)/C
```

```
30  PRINT "F="; F
50  STOP
60  DATA 9, 7, 8
70  END

BASIC IS READY
RESEQUENCE
10  REM PROGM 1
20  READ A, B, C
30  LET F= (A+B)/C
40  PRINT"F="F
50  STOP
60  DATA 9, 7, 8
70  END

BASIC IS READY

RUN

F = 2

BASIC IS READY

SAVE PROGM 1

BASIC IS READY

BYE
```

ملاحظات :

(1) حذفنا عبارة (END) رقم (40) باستخدام الأمر (DELETE) . ويمكننا كذلك حذفها بطبع رقم العبارة (40) فقط ومن ثم نضغط على المفتاح (RETURN) .

(2) صححنا العبارة (REM) رقم (10) بطبع رقم العبارة وكتابة العبارة الصحيحة ومن ثم الضغط على مفتاح (RETURN) .

(3) أقحمنا عبارة جديدة كـانت مفقودة هي عبـارة (LET) ، بطبـع رقم عبارة بين (20) و(30) وكتابة العبارة والضغط على المفتاح (RETURN) .

(4) أضفنا الى عبارة (DATA) البيـانات المتـوفرة بـطبع رقم العبـارة والكلمة الأسـاسية ومن ثم البيـانات المتـوفرة والضغط على مفتـاح (RETURN) لادخال العبارة الى الذاكرة الرئيسية للحاسبة الالكترونية .

(5) استخدمنا الأمر (LIST) للتأكد من الوضعية الجديـدة لعبارات البـرنامـج (PROGM 1) .

(6) أعدنا ترقيم عبارة البرنامج باستخـدام الأمر (RESEQUENCE) لجعلهـا منتظمة الترقيم .

(7)نفذنا البرنامج بالأمر (RUN) .

(8) خـزنا البـرنامـج في الذاكـرة الاحتياطيـة للحاسبـة الالكتـرونيـة بـالأمـر (SAVE) .

(9) أنهينا الاتصال بالحاسبة الالكترونية باستخدام الأمر (BYE) .

اختبر نفسك :

1 ـ هنالك طريقتان لحذف عبارة من برنامج بيسك. اذكرهما مع الأمثلة.

2 ـ كيف تتصرف لتصحيح أو استبدال عبارة في برنامج بيسك؟

3 ـ لماذا لا يتـم تـرقيم العبارات في بـرنامـج بيسك بفـرق رقم واحد، بـل يبدأ الترقيم بالعدد (10) كمثال، وبزيادة (10) للعبارة التالية.

4 ـ لاعــادة الــطبــع في لغــة بيســك هنــالــك أمــران همــا (LIST) ، و(CATALOG). اذكر ما هي وظيفة كل منهما .

5 ـ ما هي وظيفة الأمر (RESEQUENCE) أو (RENUMBER) ؟

7. مسائل محلولة :

سنقتبس بعض من مسائل البرمجة المحلولة في الفصل السابق لنعتاد على استخدام أوامر هذا الفصل ولنواجه تجربة تنفيذ برنامج بيسك والمفارقات التي ترافقه نتيجة لارتكاب الأخطاء أو نسيان طبع بعض المعلومات الواردة في برنامج بيسك المكتوب على ورقة بجانب المبرمج الذي يعمل على حاسبة الكترونية شخصية أو حاسبة الكترونية متوسطة أو كبيرة الحجم.

مسألة (1) : لنحاول ابتداء الاتصال بحاسبة الكترونية لخلق وتنفيذ برنامج إيجاد معدل درجات الطلاب الأربعة.

الحــــل : سنذكر هنا نفس التحاور الذي اقترحناه عند الخوض في مفردات هذا الفصل. إن معرفة تلك الاجراءات عند ابتداء الاتصال، تجعل القارىء يحسن التعامل عند إجراء ابتداء الاتصال مع حاسبتك الاتصالية، لأن تلك الاجراءات تحتوي على معظم الخطوط العامة الواجب تواجدها في أي اجراءات لابتداء الاتصال. لاحظ يمكن أن تواجه ببعض. الاختلاف بين حاسبة الكترونية وأخرى وذلك طبيعي لاختلاف أنظمة التشغيل التي تعتمد عليها تلك الحاسبات الالكترونية.

```
YOU ARE CONNECTED TO U.O.T COMPUTER SYSTEM
PLEASE TYPE YOUR ACCOUNT NUMBER
? 96854
PLEASE TYPE YOUR PASSWORD
? SAMI 1987
USER 96854 SIGNEN ON AT 12.15 HRS
WHAT LANGUAGE
? BASIC
NEW PROGRAM OR OLD
? NEW
GIVE NEW PROGRAM A NAME
? PROGM 2
BASIC IS READY

10  LET V=0
20  READ R, S, A, Z
30  LET V= (R+S+A+Z)/
40  END
50  PRINT "AVERAGE OF GRADES ="; V
60  STOP
70  DATA
30  LET V= (R+S+A+Z)/4
DELETE 40
80  END
70  DATA 75, 70, 85, 50

LIST
10  LET V=0
20  READ R, S, A, Z
30  LET V= (R+S+A+Z)/4
50  PRINT "AVERAGE OF GRADES ="; V
60  STOP
70  DATA 75, 70, 85, 50
80  END

BASIC IS READY
```

```
RUN
AVERAGE OF GRADES = 70

BASIC IS READY

02  REM COMPUTE AVERAGE OF GRADES
04  REM V= AVERAGE OF GRADES
06  REM R, S, A, Z = GRADES FOR STUDENTS
08  REM *************************
40  REM ***************************

LIST
02  REM COMPUTE AVERAGE OF GRADES
04  REM V= AVERAGE OF GRADES
06  REM R, S, A, Z = GRADES FOR STUDENTS
08  REM *********************
10  LET V=0
20  READ R, S, A, Z
30  LET V= (R+S+A+Z)/4
40  REM ************************
50  PRINT "AVERAGE OF GRADES ="; V
60  STOP
70  DATA 75, 70, 85, 50
80  END

BASIC IS READY

RUN
AVERAGE OF GRADES = 70

BASIC IS READY

RESEQUENCE

10  REM COMPUTE AVERAGE OF GRADES
20  REM V= AVERAGE OF GRADES
30  REM R, S, A, Z = GRADES FOR STUDENTS
40  REM *************************
```

```
50  LET V=0
60  READ R, S, A, Z
70  LET V= (R+S+A+Z)/4
80  REM *************************
90  PRINT "AVERAGE OF GRADES ="; V
100 STOP
110 DATA 75, 70, 85, 50
120 END

BASIC IS READY

RUN
AVERAGE OF GRADES = 70

BASIC IS READY

SAVE PROGM 2

BASIC IS READY

BYE

USER 96854 SIGNED OFF AT 12.24 HRS
TIME CONNECTED = 9 MINS
```

ملاحظـات :

(1) ابتدأنا الاتصال بالحاسبة الالكترونية بالاجراءات المعـروفة وبـدأ إدخال البرنامج عند ظهور العبارة :

BASIC IS READY

حيث تجدها تظهر عند كل مرة تنفذ بها الحاسبـة الالكترونيـة أمراً (Command) يوجه إليها من قبل الجهة المستفيدة.

(2) يتم تعتيم كلمـة السر بـالطبـع فوقهـا، ولكننـا تـركنـاهـا لتـرى من قبـل القارىء.

(3) ارتكبنا عدة أخطاء ونسينا معلومات لذا وجب علينا تنقيح البرنامج (حذف، تصحيح، استبدال، إضافة عبارة).

(4) تم إعادة كتابة البرنامج عدة مرات بواسطة الأمر (LIST) لمعرفة نتيجة التنقيح.

(5) نفذ البرنامج بالأمر (RUN).

(6) أعدنا ترقيم العبارات بانتظام باستخدام الأمر (RESEQUENCE).

(7) تم تنفيذ البرنامج بعد كل مرة يجري فيها تنقيح أو أي تعديل على عبارات البرنامج، للتأكد من أنه يؤدي الهدف الذي صمم من أجله.

(8) أنهينا الاتصال باستخدام الأمر (BYE).

مسألة (2) : إذا أردت في وقت لاحق، خزن برنامج بيسك جديد فيه أسماء بعض الأصدقاء مع أرقام هواتفهم (مسألة (2) الفصل الثالث). كيف ستنجز ذلك؟

الحل : سنحاول أولاً ابتداء الاتصال بالحاسبة الالكترونية ومن ثم نعطي اسماً لهذا البرنامج، ونجري بعض التنقيحات إذا لزم الأمر، وننفذ البرنامج لنرى النتيجة، بعد ذلك نخزن البرنامج في الذاكرة الاحتياطية للحاسبة الالكترونية، عندها ننهي الاتصال بالحاسبة الالكترونية.

```
YOU ARE CONNECTED TO U.O.T. COMPUTER SYSTEM
PLEASE TYPE YOUR ACCOUNT NUMBER
? 96854
PLEASE TYPE YOUR PASSWORD
? SAMI 1987
```

```
USER 96854 SIGNED ON AT 18.20 HRD
WHAT LANGUAGE
? BASIC
NEW PROGRAM OR OLD
? NEW
GIVE NEW PROGRAM A NAME
? PROGM 3
BASIC IS READY
10  READ A$, B$, Y, C$, Z
20  PRINT A$, X
30  PRINT B$, Y
40  PRINT C$, Z
50  STOP
60  DATA "SAMI", 8821151
70  DATA "ZEKI"
80  DATA "AHMAD", 4419867
90  END
10  READ A$, X, B$, Y, C Z
70  DATA "ZEKI", 7763210

LIST
10  READ A$, X, B$, Y, C $, Z
20  PRINT A$, X
30  PRINT B$, Y
40  PRINT C$, Z
50  STOP
60  DATA "SAMI", 8821151
70  DATA "ZEKI", 7763210
80  DATA "AHMAD", 4419867
90  END

BASIC IS READY
RUN

SAMI     8821151
ZEKI     7763210
```

```
AHMAD      4419867

BASIC IS READY
15  PRINT "NAME", "PHONE NO."
17  PRINT "----------", "---------------"
 5  REM FRIEND'S PHONE NUMBER
 LIST
 5  REM FRIEND'S PHONE NUMBER
10  READ A$, X, B$, Y, C$, Z
15  PRINT "NAME", "PHONE NO."
17  PRINT "----------", "---------------"
20  PRINT A$, X
30  PRINT B$, Y
40  PRINT C$, Z
50  STOP
60  DATA "SAMI", 8821151
70  DATA "ZEKI", 7763210
80  DATA "AHMAD", 4419867
90  END

BASIC IS READY

RUN

NAME                  PHONE NO.
------------------    --------------------------
SAMI                  8821151
ZEKI                  7763210
AHMAD                 4419687

BASIC IS READY

RESEQUENCE

BASIC IS READY

LIST

10  REM FRIEND'S PHONE NUMBER
20  READ A$, X, B$, Y, C$, Z
```

```
30  PRINT "NAME", "PHONE NO."
40  PRINT "----------", "--------------"
50  PRINT A$, X
60  PRINT B$, Y
70  PRINT C$, Z
80  STOP
90  DATA "SAMI", 8821151
100 DATA "ZEKI", 7763210
110 DATA "AHMAD", 4419867
120 END

BASIC IS READY

RUN

NAME                      PHONE NO.
--------------------------------    ----------------------------------------
SAMI                      8831151
ZEKI                      7763210
AHMAD                     4419867

BASIC IS READY

SAVE PROGM 3

BASIC IS READY

BYE

USER 96854 SIGNED OFF AT 18.27 HRS
TIME CONNECTED = 7 MINS.
```

مسألة (3) :

(أ) في احدى المرات، أردت استدعاء البرنامج (PROGM 2) وإجراء بعض التعديلات عليه. كيف ستنجز ذلك؟

(ب) إذا كنت قد خزنت في الذاكرة الاحتياطية البرامج (PROGM 1) ،

(PROGM 2) ، (PROGM 3) ، وأردت التـأكـد من ذلـك، فكيف تنجز ذلك؟

(جـ) إذا أردت حذف البرنامج (PROGM 3) من الذاكرة الاحتيـاطية. كيف ستنجز ذلك؟

الحـل : سنبـدأ الاتصـال أولاً ومن ثم نستخـدم الأمـر (OLD) لاستـدعـاء البـرنامـج (PROGM 2) وسنستخدم الأمـر (CATALOG) لادراج أسماء البـرامج المخـزونـة في الـذاكـرة الاحتيـاطيـة. أمـا حـذف البـرنـامـج (PROGM 3) من الـذاكـرة الاحتيـاطيـة، فستنجـزه بـاستخـدام الأمـر (UNSAVE) .

```
YOU ARE CONNECTED TO U.O.T. COMPUTER SYSTEM.
PLEASE TYPE YOUR ACCOUNT NUMBER
? 96854
PLEASE TYPE YOUR PASSWORD
? SAMI 1987
USER 96854 SIGNED ON AT 10.20 HRS
WHAT LANGUAGE
? BASIC
NEW PROGRAM OR OLD
? OLD PROGM 2

BASIC IS READY
LIST
10   REM COMPUTE AVERAGE OF GRADES
20   REM V= AVERAGE OF GRADES
30   REM R, S, A, Z = GRADES FOR STUDENTS
40   REM ******************
50   LET V=0
60   READ R, S, A, Z
70   LET V= (R+S+A+Z)/4
```

```
80  REM *********************
90  PRINT "AVERAGE OF GRADES ="; V
100 STOP
110 DATA 75, 70, 85, 50
120 END
```

BASIC IS READY

DELETE 10, 20, 30, 40, 80

BASIC IS READY

```
LIST
 50  LET V=0
 60  READ R, S, A, Z
 70  LET V= (R+S+A+Z)/4
 90  PRINT "AVERAGE OF GRADES ="; V
100  STOP
110  DATA 75, 70, 85, 50
120  END
```

BASIC IS READY

RESEQUENCE

BASIC IS READY

```
LIST
10  LET V=0
20  READ R, S, A, Z
30  LET V= (R+S+A+Z)/4
40  PRINT "AVERAGE OF GRADES ="; V
50  STOP
60  DATA 75, 70, 85, 50
70  END
```

BASIC IS READY

SAVE PROGM 2

```
BASIC IS READY

CATALOG

PROGM 1
PROGM 2
PROGM 3

BASIC IS READY

UNSAVE PROGM 3

BASIC IS READY

CATALOG

PROGM 1
PROGM 2

BASIC IS READY

BYE

USER 96854 SIGNED OFF AT 10.40 HRS
TIME CONNECTED = 20  MINS.
```

9. خلاصة :

يعتبر تنفيذ برنامج البيسك من مراحل حل المسألة الأكاديمية وقد أعطاها علماء الحاسبات الالكترونية (40٪) من الوقت المخصص لحل مسألة باستخدام الحاسبة الالكترونية. فحاول ممارسة استخدام الأوامر المذكورة في هذا الفصل لتستطيع تنفيذ برنامج بيسك بمهارة وتصبح العملية ممتعة أكثر منها محبطة للهمة في الاستمرار حتى استحصال النتائج المطلوبة. يمكننا تلخيص ما ذكر في هذا الفصل بالنقاط الرئيسية التالية :

أولاً : لكل حاسبة الكترونية قواعدها الخاصة لابتداء الاتصال بها. حيث يتم التحاور بين الجهة المستفيدة ونظام تشغيل تلك الحاسبة الالكترونية.

ثانياً : يستخدم الأمر (BYE) لانهاء الاتصال بالحاسبة الالكترونية.

ثالثاً : يستخدم الأمر (RUN) لتنفيذ برنامج بيسك.

رابعاً : يستخدم الأمر (NEW) لخلق برنامج جديد.

خامساً : يخزن برنامج بيسك في الذاكرة الاحتياطية لحاسبة الكترونية

باستخدام الأمر (SAVE) ويمحى البرنامج من تلك الذاكرة باستخدام الأمر (UNSAVE) .

سادساً : يمكن معرفة ما تم خزنه في الذاكرة الاحتياطية من قبل جهة مستفيدة معينة باستخدام الأمر (CATALOG) .

سابعاً : يمكن استدعاء برنامج مخزون في الذاكرة الاحتياطية باستخدام الأمر (OLD) .

ثامناً : يمكن إعادة طبع برنامج بيسك موجود في الذاكرة الرئيسية باستخدام الأمر (LIST) .

تاسعاً : يتضمن تنقيح برنامج بيسك إجراء التغييرات التالية:

(أ) تصحيح أو استبدال عبارات موجودة في البرنامج.

(ب) حذف عبارات.

(جـ) إقحام عبارة جديدة بين عبارتين.

عاشراً : تصحح العبارة بطبع رقم العبارة ومحتويات العبارة الصحيحة ومن ثم يضغط على المفتاح (RETURN) لإدخال تلك العبارة الى الذاكرة الرئيسية للحاسبة الالكترونية.

حادي عشر: تحذف العبارة باستخدام الأمر (DELETE) أو بطبع رقم العبارة ومن ثم ضغط المفتاح (RETURN) .

10. كلمات قد تعلمتها :

أوامر	COMMANDS
ابتداء الاتصال بالحاسبة الالكترونية	SIGN ON
إنهاء الاتصال بالحاسبة الالكترونية	SIGN OFF
نظام التشغيل	Operatin g System
رقم الحساب	accounr number
لغة برمجة	Language
الأمر (انهي الاتصال)	BYE
نظام المشاركة الزمنية	Time Sharing System
مفتاح موجود في لوحة مفاتيح الحاسبة الالكترونية يستخدم لإدخال عبارة من برنامج بيسك	RETURN
الأمر (جديد)	NEW
الأمر (قديم)	OLD
خطأ في قواعد نحو لغة بيسك	syntax error
خطأ في تنفيذ تعبير حسابي	execution error
خطأ في أجهزة الحاسبة الالكترونية	Hardware error
الأمر (امحو)	UNSAVE
الأمر (اخزن)	SAVE
الأمر (اعد طبع)	LIST
الأمر (ادرج البرامج)	CATALOG
الأمر (احذف)	DELETE

RESEQUENCE	الأمر (اعد تسلسل)
RENUMBER	الأمر (اعد ترقيم)
Editing	تنقيح
Terminal	جهاز طرفي

11. تمارين (4) :

استخدم كلمات قد تعلمتها .

(أ) بنظام المشاركة الزمنية .

(ب) لغة البرمجة .

(جـ) نظام تشغيل .

(د) إنهاء الاتصال بالحاسبة الالكترونية .

(هـ) لابتداء الاتصال بالحاسبة الالكترونية .

(و) المفتاح (RETURN) .

(ز) أوامر .

(ح) تنقيح برنامج بيسك .

(ط) خطأ في قواعد لغة بيسك .

(ى) رقم الحساب .

1 ـ لكل حاسبة الكترونية قواعدها الخاصة بها ________ .

2 ـ لإدخال عبارة من برنامج بيسك في الذاكرة الرئيسية للحاسبة الالكترونية يجب الضغط على ________ .

3 ـ تستطيع أكثر من جهة مستفيدة استخدام نفس الحاسبة الالكترونية في وقت واحد، تدعى هذه الطريقة ________ .

4 ـ يتعرف نظام التشغيل على الجهة المستفيدة من ــــــــــ .

5 ـ تستطيع ــــــــــــــ باستخدام الأمر (BYE) .

6 ـ تتحاور الجهة المستفيدة مع ــــــــ الحاسبة الالكترونية لتنفيذ برنامج بلغة بيسك .

7 ـ أحد الأسئلة التي يريد نظام التشغيل معرفتها من الجهة المستفيدة هو نوعية ــــــــ المستخدمة .

8 ـ تتوقف الحاسبة الالكترونية عن تنفيذ برنامج بيسك إذا حدث ــــــــــ .

9 ـ عند ــــــــ يتم تصحيح أو استبدال أو حذف أو اقحام عبارات في برنامج بيسك .

10 ـ تستطيع تنفيذ برنامج البيسك بواسطة حاسبة الكترونية إذا تعلمت وظائف ــــــ لغة بيسك الخاصة بتلك الحاسبة .

صح أم خطأ :

1 ـ تتبع جميع الحاسبات الالكترونية نفس قواعد ابتداء الاتصال بالحاسبة الالكترونية .

2 ـ يستخدم المفتاح (RETURN) لاظهار عبارة في برنامج بيسك .

3 ـ لا تخفى كلمة سر الجهة المستفيدة من قبل الحاسبة الالكترونية بل تترك كما هي .

4 ـ تجبر إجراءات الاتصال بالحاسبة الالكترونية على أن يكون أول أمر هو (SAVE) .

5 ـ تستمر الحاسبة الالكترونية في تنفيذ برنامج بيسك حتى لو حدث خطأ في أجهزتها.

6 ـ يتم استدعاء برنامج بيسك مخزون في الذاكرة الاحتياطية باستخدام الأمر (CATALOG) .

7 ـ يستخدم الأمر (DELETE) لاقحام عبارة جديدة بين عبارتين.

8 ـ تصحح العبارة في برنامج بيسك باستخدام الأمر (RESEQUENCE) .

9 ـ يمكن استدعاء برنامج موجود في الذاكرة الرئيسية للحاسبة الالكترونية باستخدام الأمر (LIST) .

10 ـ تحذف العبارة في برنامج بيسك بطبع رقم العبارة ومن ثم الضغط على المفتاح (RETURN) .

أجب باختصار :

1 ـ لماذا يكون هنالك فرق بين إجراءات ابتداء الاتصال بالحاسبة الالكترونية الشخصية وحاسبة الكترونية كبيرة الحجم؟ اذكر ذلك الفرق.

2 ـ كيف ينهى الاتصال بالحاسبة الالكترونية؟

3 ـ كيف يتم إدخال عبارة في برنامج بيسك الى الذاكرة الرئيسية بعد أن تطبع على جهاز إدخال البيانات؟

4 ـ اذكر الأمر الذي يستخدم لأداء كل وظيفة مما يلي :

(أ) استدعاء برنامج قديم.

(ب) استدعاء أسماء برامج جهة مستفيدة.

5 ـ ما هي التغيرات التي يمكن اجراؤها عند تنقيح برنامج بيسك؟

6 ـ كيف يتم تصحيح عبارة في برنامج بيسك بعد أن تم خزنها في الذاكرة الرئيسية؟

7 ـ ما هي الأمور التي يمكن أن توقف حاسبة الكترونية من تنفيذ برنامج بيسك؟

8 ـ ما هي وظيفة كل أمر مما يلي :

(أ) الأمر (NEW) .

(ب) الأمر (OLD) .

9 ـ ما هي نتائج خزن برنامج بيسك في الذاكرة الاحتياطية؟ وكيف يخزن ذلك البرنامج؟ اذكر أمراً لمحو ذلك البرنامج .

10 ـ اذكر اسمين لأمر يمكن استخدامه لإعادة ترقيم عبارات برنامج بيسك .

اختار الجواب الصحيح :

1 ـ يعمل الأمر (BYE) على :

(أ) ابتداء الاتصال بالحاسبة الالكترونية .

(ب) تنقيح عبارات برنامج بيسك .

(جـ) خزن برنامج بيسك .

(د) إنهاء الاتصال بالحاسبة الالكترونية .

2 ـ تكون وظيفة الأمر (SAVE) هي :

(أ) ابتداء الاتصال بالحاسبة الالكترونية .

(ب) تنقيح عبارات برنامج بيسك.
(جـ) خزن برنامج بيسك.
(د) إنهاء الاتصال بالحاسبة الالكترونية.

3 ـ يتم حذف عبارة في برنامج بيسك ________
(أ) بواسطة الأمر (NEW).
(ب) بواسطة الأمر (OLD) .
(جـ) بواسطة الأمر (DELETE) .
(د) بواسطة الأمر (LIST).

4 ـ بواسطة أوامر (Commands) لغة بيسك ______
(أ) يمكنك كتابة برنامج بيسك.
(ب) يمكنك تنفيذ وتصحيح برنامج بيسك على حاسبة.
(جـ) يمكنك كتابة خوارزمية حل.
(د) يمكنك قراءة برنامج بيسك مكتوبة على ورقة.

5 ـ يتم خلق برنامج بيسك ______
(أ) بواسطة الأمر (NEW) .
(ب) بواسطة الأمر (OLD) .
(جـ) بواسطة الأمر (DELETE) .
(د) بواسطة الأمر (LIST) .

6 ـ يتم إعادة طبع برنامج بيسك موجود في الذاكرة الرئيسية:
(أ) بواسطة الأمر (NEW) .
(ب) بواسطة الأمر (OLD) .
(جـ) بواسطة الأمر (DELETE) .

(د) بواسطة الأمر (LIST) .

7 ـ ينفّذ برنامج بيسك ______

(أ) بواسطة الأمر (CATALOG) .

(ب) بواسطة الأمر (RESEQUENCE) .

(جـ) بواسطة الأمر (RUN) .

(د) بواسطة الأمر (RETURN) .

8 ـ يتم إعادة ترقيم عبارات برنامج بيسك مرقم بصورة غير منتظمة :

(أ) بواسطة الأمر (CATALOG) .

(ب) بواسطة الأمر (RESEQUENCE) .

(جـ) بواسطة الأمر (RUN) .

(د) بواسطة الأمر (RETURN) .

9 ـ يتم استدعاء جميع البرامج المخزونة في الذاكرة الاحتياطية والخاصة بجهة مستفيدة معينة :

(أ) بواسطة الأمر (CATALOG) .

(ب) بواسطة الأمر (RESEQUENCE) .

(جـ) بواسطة الأمر (RUN) .

(د) بواسطة الأمر (RETURN) .

10 - تخزن عبارة في برنامج بيسك بعد أن يتم طبعها على جهاز ادخال البيانات :

(أ) بواسطة الأمر (CATALOG)

(ب) بواسطة الأمر (RUN)

(جـ) بواسطة الأمر (RESEQUENCE)

(د) بالضغط على المفتاح (RETURN)

الفصل الخامس

شروط.. وانتقال... وتكرار
ومن ثم اتخاذ قرار

1. مقدمة
2. الخروج عن التسلسل الاعتيادي لعبارات البرنامج
3. الخروج المشروط عن التسلسل الاعتيادي لعبارات البرنامج
   اختبر نفسك
4. الانتقال الانتقائي الى احد عبارات البرنامج
5. كيفية السيطرة على تكرار انشوطة؟
   اختبر نفسك
6. المصفوفات ـ احدى تطبيقات انشوطة عبارتي (FOR/NEXT)
7. وضع برنامج صغير للعمل داخل برنامج كبير
   اختبر نفسك
8. مسائل برمجة محلولة
9. خلاصة
10. كلمات قد تعلمتها
11. تمارين (5)

1. مقدمة :

ما يميز الحاسبة الالكترونية عن أجهزة الاحتساب الحديثة هو امكانياتها على إجراء الحسابات والعمليات الرياضية بصورة متكررة. ويمكن السيطرة على عدد المرات التي تنفذ بها تلك العمليات عن طريق اختبار تحقق شرط معين ضمن خطوات انشوطة، عندها يتخذ القرار للانتقال الى تنفيذ سلسلة جديدة من عبارات برنامج البيسك.

توجد عبارات في لغة بيسك تستخدم لتكوين انشوطة والسيطرة عليها. وستكون من ضمن أهداف هذا الفصل، بصورة عامة، شرح تلك العبارات. إضافة لذلك سنتطرق الى عبارات السيطرة في لغة بيسك. سيكون بإمكان القارىء بعد انتهائه من دراسة مفردات هذا الفصل:

1 ـ معرفة واستخدام عبارات تكوين الانشوطة في لغة بيسك.

2 ـ معرفة واستخدام عبارات الانتقال المشروط وغير المشروط والانتقائي في لغة بيسك.

3 ـ معرفة واستخدام أساليب السيطرة على تكرار انشوطة.

4 ـ معرفة واستخدام المصفوفات.

5 ـ كيفية كتابة برنامج فرعي للعمل داخل برنامج رئيسي.

2. الخروج عن التسلسل الاعتيادي لعبارات برنامج بيسك :

ذكرنا في تركيب برنامج بيسك أن تنفيذ عبارات برنامج بيسك يتم بالتسلسل من أدنى رقم عبارة إلى أن تنفذ الحاسبة الالكترونية عبارة (END) . هذا صحيح. ولكن هنالك عبارة في لغة بيسك تنقل وقائع التنفيذ الى عبارة أخرى متجاوزة تسلسل العبارات في برنامج البيسك. تلك هي عبارة (GOTO) أي (اذهب إلى).

تتركب عبارة (GOTO) من رقم العبارة والكلمة الأساسية (GOTO) ومن رقم عبارة يتم الانتقال إليها بدون شرط أو قيد. يمكنك كتابة (GOTO) كلمة واحدة متلاصقة أو يمكنك الفصل بينهما بفراغ.

```
 80   GO TO 30
170   GOTO 210
```

تستطيع عبارة (GOTO) القفز إلى رقم عبارة أعلى أو أدنى من رقمها هي. أي يمكنها نقل تنفيذ العبارات إلى نهاية البرنامج أو بدايته.

مثال : سنعيد كتابة برنامج معدّل الطلاب الأربعة بحيث نجعله أبدي التنفيذ لا يتوقف إلا بالتدخل من قبل المشغل إذا كنت تجري حساباتك على حاسبة الكترونية كبيرة أو بالضغط على المفتاح

(PREAK) الموجود في لوحة مفاتيح حاسبتك الالكترونية الشخصية.

```
10  LET V=0
20  INPUT R, S, A, Z
30  LET V= (R+S+A+Z)/4
40  PRINT "AVERAGE OF GRADES ="; V
50  GO TO 10
60  STOP
70  END

RUN
? 10, 20, 30, 60

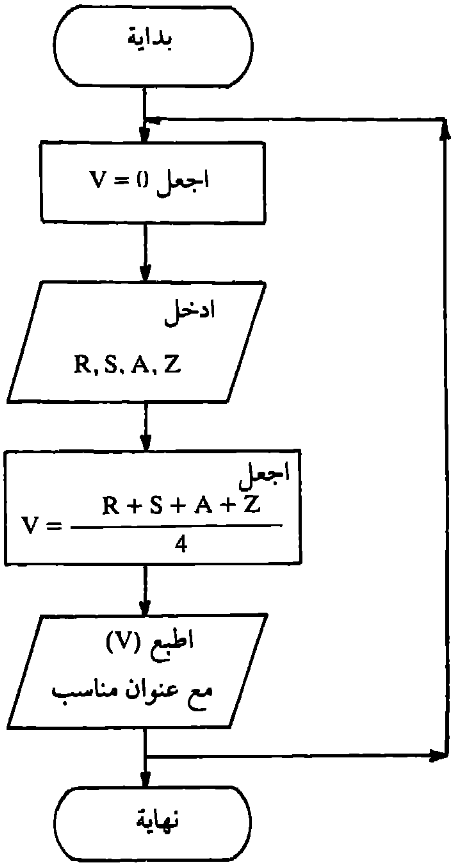
AVERAGE OF GRADES = 30

? 40, 50, 70, 80

AVERAGE OF GRADES = 60
```

وهكذا سوف يستمر طلب الحاسبة الالكترونية لبيانات تجرى عليها الحسابات وتخرج النتيجة، ومن ثم تطلب إدخال بيانات جديدة. يمكنك الخروج من دائرة تكرار التنفيذ بالضغط على مفتاح (PREAK) أو أية وسيلة أخرى تتيحها لك حاسبتك الالكترونية لإيقاف تنفيذ برنامج.

وبذلك استخدمنا عبارة (GOTO) لنقل تنفيذ العبارات غير المشروط في برنامج بيسك. حيث نجد عبارة (GOTO) تستخدم لتكوين انشوطة. وتمثل عبارة (GOTO) في المخطط الانسيابي بخط انسيابي يربط بين صندوقين. لنرسم المخطط الانسيابي لمثال إيجاد معدل الدرجات المذكور أعلاه بعد إدخال عبارة (GOTO) عليه.


بداية
اجعل V = 0
ادخل
R, S, A, Z
اجعل
$V = \frac{R + S + A + Z}{4}$
اطبع (V)
مع عنوان مناسب
نهاية

شكل (1 . 5)

3. الخروج المشروط عن التسلسل الاعتيادي لعبارات برنامج بيسك :

من المفيد جداً في تطبيقات كثيرة أن ينتقل تسلسل تنفيذ العبارات إلى موقع آخر من التسلسل الاعتيادي نتيجة لتحقق شرط معين. في لغة بيسك، يمكنك وضع الشرط ورقم العبارة التي سيتم الانتقال إليها عند تحقق الشرط في عبارة (IF-THEN) والتي تعني (إذا كان ـ فإن).

تتركب عبارة (IF - THEN) من رقم العبارة والكلمة الأساسية (IF) يأتي بعدها الشرط مصاغ بتعبير متراجحة (RELATIONAL EXPRESSION)، ومن ثم تأتي الكلمة الأساسية (THEN) لتدل على رقم العبارة التي سيتم الانتقال إليها عندما يكون تعبير المتراجحة متحققاً (TRUE).

يمكنك مراجعة تركيب تعبير المتراجحة في الفصل الثاني الفقرة (6) من هذا الكتاب. حيث تجد أن تعبير المتراجحة يتألف من تعبيرين حسابيين أو تعبيرين مقطعيين يربط بينهما عملية مقارنة. توجد ستة عمليات مقارنة.

أمثلة :

```
90  IF ABS (X−Y) < = 0.001 THEN 50
20  IF 3+INT (X ↑3 ★Y ↑ 3) > = 99 THEN 80
40  IF A$ = "YES" THEN 190
```

```
80   IF "BASIC" < > A$THEN 20
160  IF A$ = B$THEN 305
```

إذا لم يتحقق شرط عبارة (IF-THEN) تستمر الحاسبة الالكترونية في تنفيذ العبارة التي تلي عبارة (IF-THEN) مباشرة سنوضح ذلك بالمثال البسيط التالي :

مثال : سنجري أسلوب التنفيذ الجاف على عبارة (IF - THEN) التالية :

```
40   IF A * 10 > = B/C THEN 230
```

A	B	C	تعبير المتراجحة	تحقق الشرط
1	4	2	10 > = 2	(1) نعم (TRUE)
2	100	4	20 > = 25	(2) كلا (FALSE)
30	90	3	30 > = 30	(3) نعم (TRUE)

وبذلك سينتقل تنفيذ عبارات البرنامج إلى العبارة رقم (230) نتيجة لتحقق الشرط في المثالين الأول والثالث. بينما ستستمر الحاسبة الالكترونية في تنفيذ العبارة التي تلي عبارة (IF - THEN) رقم (40) نتيجة لعدم تحقق الشرط في تعبير المتراجحة.

مثال : سنأخذ بنظر الاعتبار في هذا المثال كون التعبيرين مقطعيين، ونختبر وظيفة عبارة (IF - THEN) .

```
90   IF A$ = "YES" THEN 20
```

السبب	تحقق الشرط	تعبير المتراجحة	A$
وجود فراغ قبل الحرف (Y) مما يجعل قيمة المتغير المقطعي (A$) تختلف عن قيمة الثابت المقطعي على يمين تعبير المتراجحة المقطعي.	كلا	"YES"="YES"	"YES"

إذن يختبر تحقق شرط معين باستخدام عبارة (IF - THEN) ليتم الانتقال إلى العبارة التي كتب رقمها بعد الكلمة الأساسية (THEN). أما إذا لم يتحقق فستنفذ العبارة التي تلي عبارة (IF - THEN) مباشرة.

تمثل عبارة (IF - THEN) في المخطط الانسيابي برمز القرار المعيني (◇)، حيث تخرج من هذا الرمز خطي انسياب أحدهما يكتب عليه الكلمة (نعم) والآخر يكتب عليه (كلا)، وهما على التعاقب، دلالة على اتجاه تنفيذ خطوات الحل في حالة تحقق أو عدم تحقق الشرط الموجود داخل الرمز المعيني.

قبل ذكر الأمثلة سنعطي طريقتين للسيطرة على تكرار تنفيذ انشوطة. الأولى تتم باستخدام العدّاد (counter)، والثانية باستخدام قيمة ملحقة أو ذيلية (Trailer value) .

العداد هو متغير رقمي يستخدم لحساب عدد المرات التي قد نفذت بها الانشوطة. عند استخدام العداد يجب إجراء التالي:

(أ) تمهيد العداد لإعطائه قيمة أولية عند بدء التكرار.

(ب) زيادة قيمة العداد في كل مرة تنفذ فيها الانشوطة.

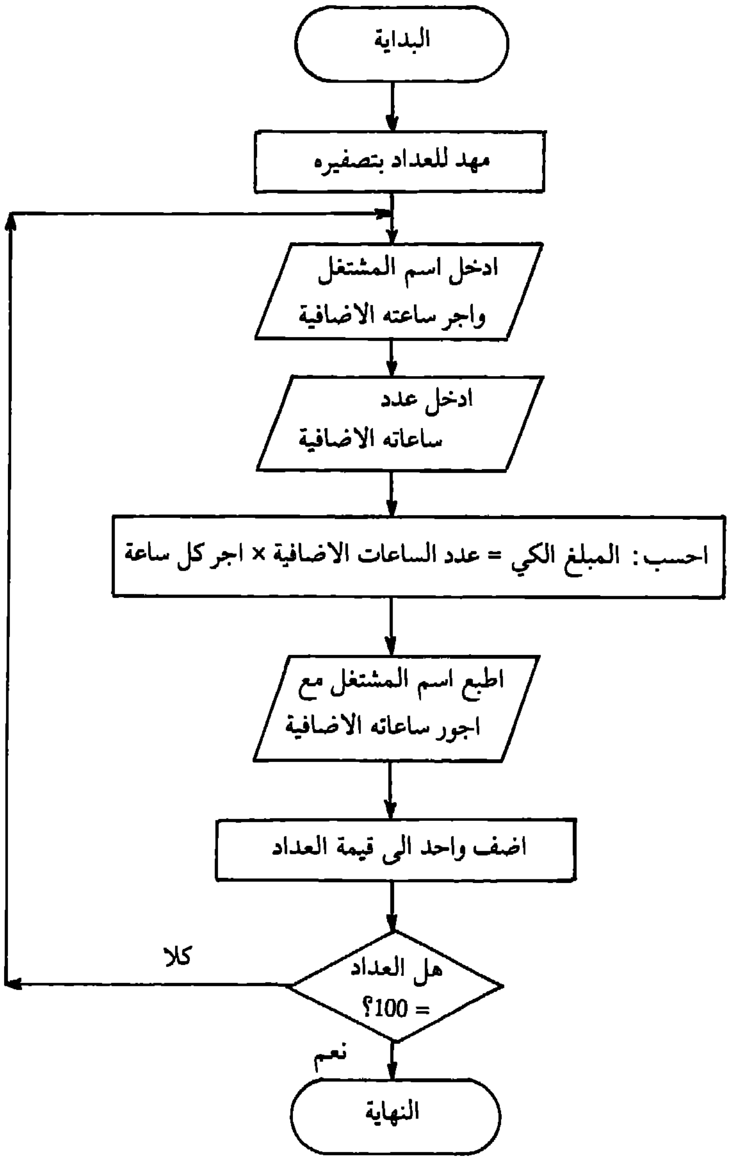
(جـ) اختبار قيمة العداد لمعرفة هل بلغت المرات التي نفذت بها

الانشوطة العدد المطلوب؟ يتم ذلك باستخدام عبارة (IF - THEN) .

أما القيمة الملحقة أو الذيلية فهي قيمة رقمية توضع في نهاية قائمة البيانات لتؤشر الوصول إلى نهاية تلك البيانات. تستخدم عبارة (IF - THEN) لاختبار الوصول الى القيمة الملحقة.

مثال: هذا المثال يجسد أسلوب العداد للسيطرة على تكرار تنفيذ انشوطة. ولهذا المثال نفس فكرة المسألة رقم (1) من المسائل المحلولة في الفصل الأول ما عدا بعض التحويرات التي تناسب مرحلة القارىء المتقدمة للبرمجة بلغة بيسك حيث تجد المخطط الانسيابي أدناه هو تجسيد لخطوات حل، كيفية إيجاد أجور الساعات الإضافية لمائة مشتغل في معمل للحاسبات الالكترونية. تحسب الأجور الإضافية من ضرب عدد الساعات الإضافية في أجر كل ساعة.

أصبحت عملية نقل خطوات خوارزمية إلى عبارات برنامج بيسك بسيطة لدى القارىء، لذلك سنكتب العبارات مباشرة باتباع خطوات المخطط الانسيابي. بحيث سندخل البيانات باستخدام عبارة (INPUT) أو عبارتي (DATA + READ)، ونجري الحسابات باستخدام عبارة (LET)، ونخرج النتائج باستخدام العبارة (PRINT). وسنسيطر على تكرار تنفيذ الانشوطة، التي تحتوي الحسابات، باختبار العداد باستخدام العبارة (IF - THEN)، أما العبارة (GOTO) فستكوّن الانشوطة.


البداية
مهد للعداد بتصفيره
ادخل اسم المشتغل
واجر ساعته الاضافية
ادخل عدد
ساعاته الاضافية
احسب: المبلغ الكي = عدد الساعات الاضافية × اجر كل ساعة
اطبع اسم المشتغل مع
اجور ساعاته الاضافية
اضف واحد الى قيمة العداد
هل العداد
= 100؟
كلا
نعم
النهاية

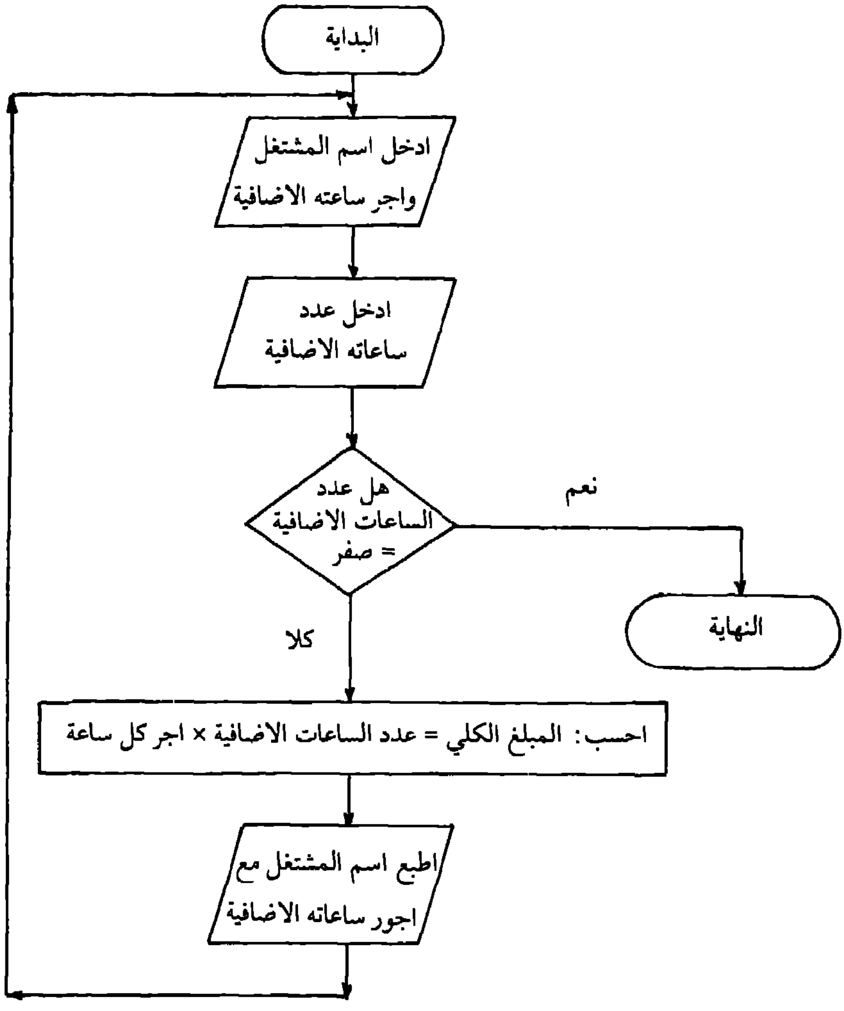
شكل (2 . 5)

```
10  REM AN EXAMPLE FOR THE COUNTER
20  REM C=COUNTER
30  REM N$ = WORKER'S NAME, H = HOURLY RATE
40  REM W = WORKED HOURS
50  REM T = OVERTIME WAGE
60  LET C=0
70  PRINT "WORKER'S NAME (N$), HOURLY RATE (H)"
80  INPUT N$ , H
90  PRINT "WORKED HOURS (W) FOR"; N$
100 INPUT W
110 LET T= H * W
120 PRINT "WORKER'S NAME", "OVERTIME WAGE"
130 PRINT "---------------  ",  "---------------"
140 PRINT N$ , T
150 LET C= C+1
160 IF C=100 THEN 180
170 GO TO 70
180 STOP
190 END
```

لاحظ ارتبـاط عبارة (IF - THEN) وعبـارة (GOTO) لتكوين انشـوطـة والسيطرة على تكرار الانشوطة باستخدام فكـرة العداد. الآن نفـذ البرنـامج أعـلاه فستجده يتحـاور معك كمـا لو كـانت الحاسبـة الالكترونيـة تجيد فن المحادثة.

مثـال: هذا المثـال يجسد فكـرة القيمة الملحقـة أو الـذيليـة للسيـطرة على تكرار تنفيذ انشـوطة. وسنقتبس فكـرة هذا المثـال من المسألـة رقم (3) من المسـائـل المحلولـة في الفصـل الأول. حيث جعلنـا عـدد العاملين في مصنع الحـاسبات الالكتـرونية غيـر معلوم. واستخدمنـا الصفـر كقيمة ملحقـة للدلالة على الـوصول إلى نهـايـة البيـانـات.

سنجعل القيمة الملحقة عند المتغير الرقمي الخاص بعدد الساعات الإضافية.



شكل (3 . 5)

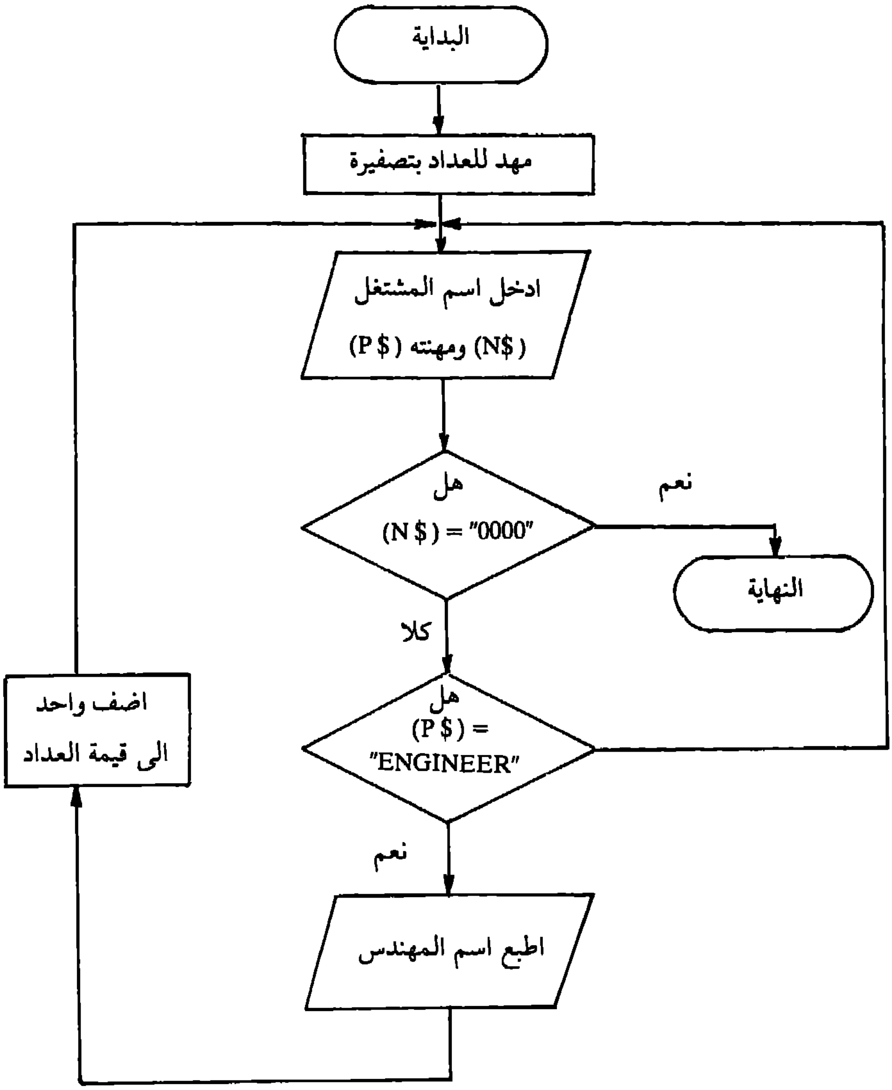
سنوضح المتغيرات المستخدمة في البرنامج باستخدام العبارة (REM). وسندخل البيانات باستخدام العبارة (INPUT) مسبوقة بعبارة (PRINT) لإرشاد المستخدم للبرنامج على نوعية البيانات التي يجب إدخالها. وسنستخدم العبارة (LET) لاجراء الحسابات، والعبارة (PRINT) لإخراج النتائج مع عناوين مناسبة. ولتكوين الانشوطة نستخدم العبارة (GOTO) وللسيطرة عليها فسنختبر عدد الساعات الإضافية هل هي مساوية للقيمة الملحقة (صفر) باستخدام العبارة (IF - THEN) .

```
10  REM AN EXAMPLE FOR THE TRAILER VALUE
20  REM TRAILING VALUE = W=0
30  REM N$ = WORKER'S NAME, H = HOURLY RATE
40  REM W= WORKED HOURS
50  REM T= OVERTIME WAGE
60  PRINT "WORKER'S NAME (N$), HOURLY RATE (H)"
70  INPUT N$ , H
80  PRINT "WORKED HOURS (W) FOR"; N$
90  INPUT W
100 IF W=0 THEN 160
110 LET T= H * W
120 PRINT "WORKER'S NAME (N$)", "OVERTIME WAGE (T)"
130 PRINT "---------------", "---------------"
140 PRINT N$ , T
150 GO TO 60
160 STOP
170 END
```

نفّذ البرنامج أعلاه باستخدام حاسبتك الالكترونية.

مثال: سنوضح في هذا المثال كيفية استخدام العبارة (IF - THEN) لاختبار شرط يتكون من تعبير مقطعي. حيث سيحصى عدد المهندسين في مصنع الحاسبات الالكترونية. علماً أننا سنجعل قيمة ملحقة أو ذيلية عند نهاية قائمة أسماء المشتغلين. وستكون القيمة الملحقة مساوية إلى "0000" .

```
10  REM COMPUTING NUMBER OF ENGINEERS
20  REM N$ = WORKER'S NAME
30  REM P$ = WORKER'S OCCUPATION
40  REM C = COUNTER
50  LET C=0
60  PRINT "WORKER'S NAME (N$), HIS OCCUPATION (P$)"
70  INPUT N$, P$
80  IF N$ = "0000" THEN 140
90  IF P$ = "ENGINEER" THEN 110
100 GO TO 60
110 PRINT N$ ; "IS AN ENGINEER"
120 LET C= C+1
130 GO TO 60
140 PRINT "THE NUMBER OF ENGINEERS ="; C
150 STOP
160 END
```


البداية
مهد للعداد بتصفيرة
ادخل اسم المشتغل
(N$) ومهنته (P $)
هل
(N $) = "0000"
نعم
النهاية
كلا
هل
(P $) =
"ENGINEER"
نعم
اطبع اسم المهندس
اضف واحد
الى قيمة العداد

شكل (4 . 5)

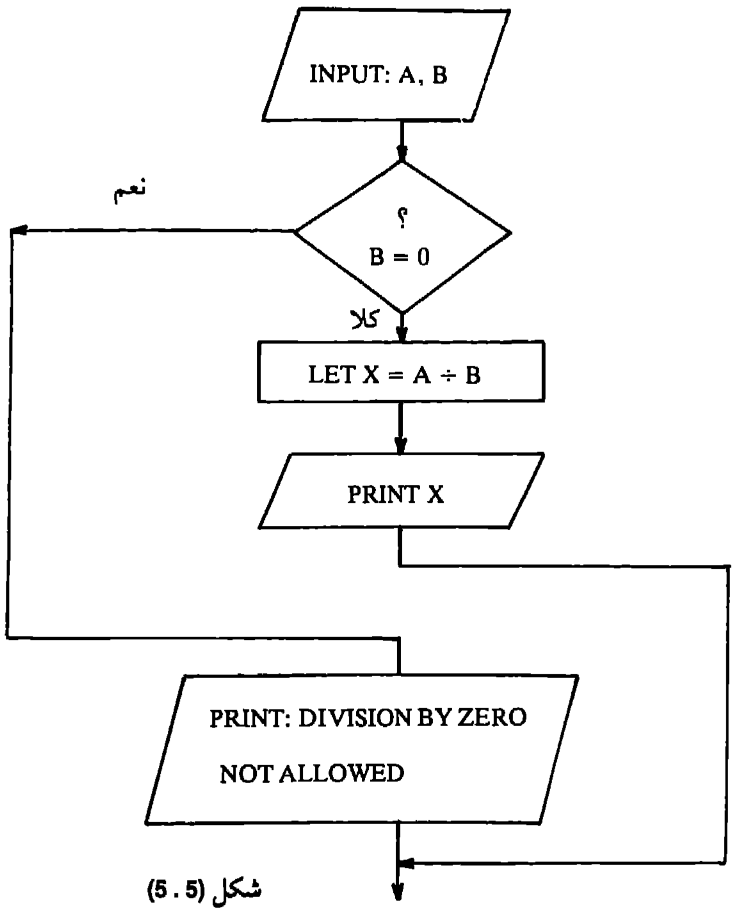
اختبر نفسك :

1 ـ هل سيتم الانتقال إلى عبارة رقم (90)

```
20   IF 5 ↑ (X * Y) > = 26 THEN 90
```

إذا كان (Y=1) و (X=2) .

2 ـ اكتب جزء من برنامج بيسك الذي ينقل فكرة الخوارزمية التالية:



شكل (5 . 5)

3 ـ اذكر عبارة في لغة بيسك تستخدم لتكوين انشوطة؟

4 ـ ما الفرق بين الانتقال الذي تحدثه عبارة (GOTO) عن ذلك الذي تحدثه عبارة (IF - THEN) .

5 ـ كيف يمكن السيطرة على تكرار تنفيذ انشوطة؟

4. الانتقال الانتقائي إلى أحد عبارات البرنامج :

عرفت نوعين من الانتقال في برنامج بيسك، الأول: انتقال غير مشروط باستخدام عبارة (GOTO). والثاني : انتقال مشروط باستخدام عبارة (IF - THEN). حيث يعتمد الانتقال المشروط على أن الجواب هو (نعم) أو (كلا) ولكن ما هي العبارة المناسبة في لغة بيسك لاتخاذ قرار له اختيارات أكثر من (نعم) أو (كلا)؟ توجد في لغة بيسك عبارة (ON/GOTO) تمكننا من اتخاذ قرار في حالة وجود اختيارات كثيرة. ومن التطبيقات المهمة لهذه العبارة هو إظهار قائمة اختيارات (menu) للمستفيد من الحاسبة الالكترونية، أو تصميم برامج لاختبار معلومات المستفيد.

تتركب عبارة (ON/GOTO) من رقم العبارة والكلمة الأساسية (ON) تتبع بمتغير رقمي أو تعبير حسابي ومن ثم الكلمة الأساسية(GOTO)تتبع بأرقام عبارات التي سيتم انتقاء أحدها ليتم تنفيذها.

مثال :

```
55   ON A*B GO TO 100, 140, 280, 20, 500
```

عند تنفيذ عبارة (ON/GOTO) يتم إيجاد قيمة التعبير الحسابي فإذا

كانت النتيجة عدد عشري (decimal) فإن جزءه الصحيح فقط يؤخذ بنظر الاعتبار. سيسبب تنفيذ العبارة اعلاه ما يلي :

يتم الانتقال إلى عبارة (100) إذا كانت A*B=1

(140) إذا كانت A*B=2

(280) إذا كانت A*B=3

(020) إذا كانت A*B=4

(500) إذا كانت A*B=5

يجب أن تقع قيمة التعبير الحسابي اعلاه في المدى (1 - 5) وإلا يتم تخطي عبارة (ON/GOTO) الى العبارة التي تليها في البرنامج. يمكنك الانتقال الى عدد غير محدد من العبارات، إلا بإمكانية استيعاب أرقام العبارات في عبارة (ON/GOTO) .

مثال: في تمارين كل من الفصول الأربعة السابقة، قد تم اختباران بأربعة أنواع من الاختبارات في كل تمرين. كان أحدهما هو «اختار الجواب الصحيح». نستطيع جعل اختبارات التمارين تتم باستخدام الحاسبة الالكترونية كممتحن للطالب. وكمثال على ذلك سنبرمج السؤال السادس من اختبار الفصل الأول باستخدام عبارة (ON/GOTO) في لغة بيسك. ينص السؤال:

* قسّم أساتذة البرمجة الوقت المحدد لحل مسألة باستخدام الحاسبة الالكترونية. وقد خصصوا لكتابة الخوارزمية جزء قدره ـــــــــ .

أ - (80٪)

ب - (60٪)

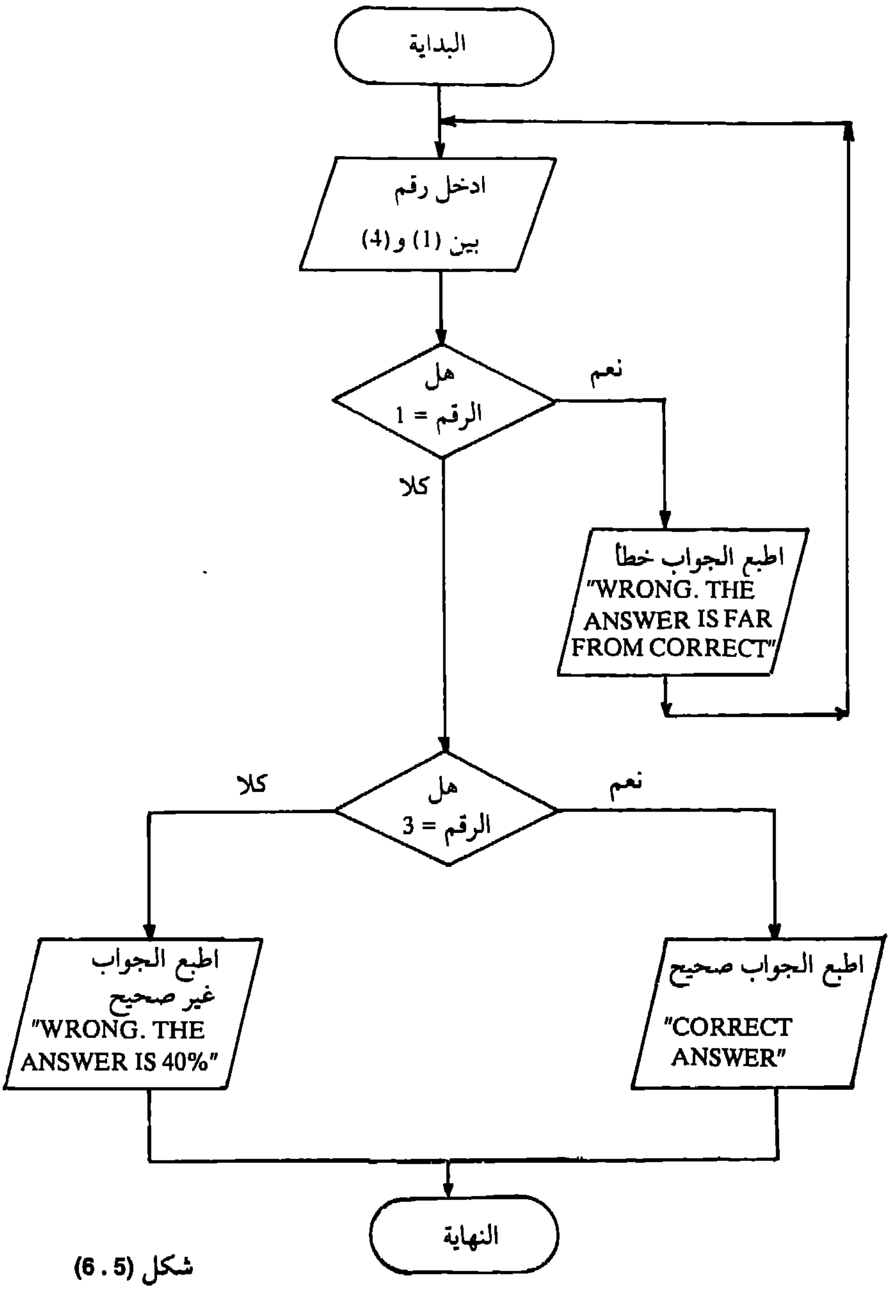
جـ - (40٪)

د - (20٪)

بحيث نجعل السؤال يظهر للطالب كما يلي :

```
WHAT PERCENTAGE OF TIME IS DEDICATED
TO ALGORITHM WRITING ? IS IT :
1. 80%
2. 60%
3. 40%
4. 20%
GIVE YOUR ANSWER BY TYPING THE
CORRESPONDING NUMBER (1 - 4) : □
```

وتعطى درجة (10) للجواب الصحيح أي الفرع (3). ولا تعطى درجة للجواب غير الصحيح . بحيث يتم تنبيه الطالب الى أن جوابه خاطىء جداً في حالة طبعه الرقم (1) . سنقترح عليك المخطط الانسيابي في شكل (5 . 6) كخوارزمية للحل. بعد ذلك تصبح كتابة البرنامج بسيطة باتباع انسياب خطوات خوارزمية الحل ونقلها على شكل عبارة من لغة بيسك لإدخال البيانات ومعالجتها ومن ثم طبع النتائج مع العنوان المناسب.


البداية
ادخل رقم
بين (1) و(4)
هل
الرقم = 1
نعم
كلا
اطبع الجواب خطأ
"WRONG. THE ANSWER IS FAR FROM CORRECT"
هل
الرقم = 3
كلا
نعم
اطبع الجواب
غير صحيح
"WRONG. THE ANSWER IS 40%"
اطبع الجواب صحيح
"CORRECT ANSWER"
النهاية

شكل (6.5)

```
10  REM A MULTIPLE CHOICE TEST
20  PRINT "WHAT PERCENTAGE OF THE TIME IS
    DEDICATED"
30  PRINT "TO ALGORITHM WRITING?";
40  PRINT "IS IT :"
50  PRINT "1. 80%"
60  PRINT "2. 60%"
70  PRINT "3. 40%"
80  PRINT "4. 20%"
90  PRINT "GIVE YOUR ANSWER BY TYPING THE"
100 PRINT "CORRESPONDING NUMBER (1 - 4):    "
110 PRINT A
120 ON A GO TO 140, 150, 130, 150
130 PRINT "CORRECT ANSWER"
140 PRINT "WRONG . THE ANSWER IS FAR FROM CORRECT"
150 PRINT "WRONG . THE ANSWER IS (40%)."
160 STOP
170 END
135 GO TO 160
145 GO TO 90
```

لو طلبت من الحاسبة الالكترونية أن تسّطر (LIST) عبـارات البرنـامج اعـلاه لجعلت آخـر عبـارتين (135) و(145) في مكـانهـا المتسلسـل بين عبارات برنامج بيسك. ولو نفذت البرنامج اعلاه لظهـرت لك النتـائج كمـا تم التخطيط لذلك في المخطط الانسيابي اعلاه.

يمكننا الآن تلخيص ما جـاء بخصوص العبـارة (ON/GOTO)، حيث تنقل هذه العبارة التنفيذ الى عبـارة من مجموعـة عبارات معتمـدة على قيمة تعبير حسابي أو متغير رقمي.

(أ) يعتبر هذا الانتقال انتقائي شرطي

(ب) إذا كانت نتيجة التعبير الحسابي عدد عشري فإن الحاسبة الالكترونية تأخذ العدد الصحيح فقط من العدد الكلي.

(جـ) لا تنفذ عبارة (ON/GOTO) إذا كانت نتيجة التعبير الحسابي أو قيمة المتغير الرقمي صفراً أو عدداً صحيحاً أكبر من عدد أرقام العبارات المتوفرة.

(د) من تطبيقات هذه العبارة هو اختبار معلومات المستفيد باختبار «اختار الجواب الصحيح» أو بتوفير قائمة من الاختيارات المقترح إجرائها من قبل الحاسبة الالكترونية للجهة المستفيدة.

(هـ) يعبر عن عبارة (ON/GOTO) بأكثر من رمز قرار معيني في المخطط الانسيابي لخوارزمية حل مسألة.

5. كيفية السيطرة على تكرار انشوطة ؟

توجد في لغة بيسك ثلاثة أساليب للسيطرة على تكرار انشوطة:

1 ـ باستخدام العداد (counter) .

2 ـ باستخدام القيمة الملحقة أو الذيلية (trailer value) .

3 ـ باستخدام عبارتي (FOR/NEXT) .

قد تعرّفت على أسلوبي العداد والقيمة الملحقة في الفقرة رقم (3) من هذا الفصل. أما الأسلوب الثالث الذي يستخدم عبارتي (FOR) و(NEXT) لتكوين انشوطة والسيطرة عليها، فسيكرس له هذه الفقرة.

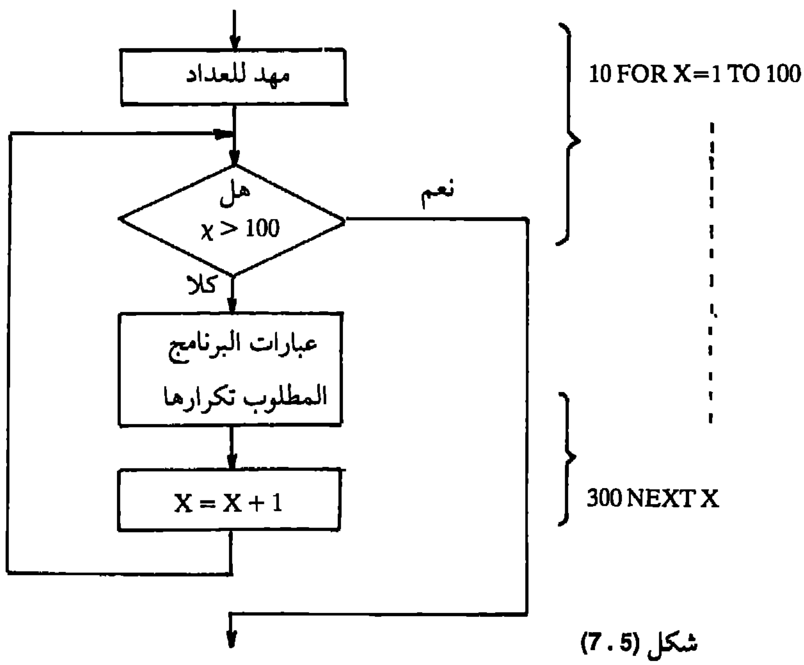
تكوين العبارة (FOR) أول عبارة في الانشوطة، ويتم بواسطتها ذكر عدد المرات التي ستنفذ بها الانشوطة. أما عبارة (NEXT)، فعندها تنتهي الانشوطة، وينقل بواسطتها تنفيذ العبارات الى أول عبارة في الانشوطة.

تتركب عبارة (FOR) من رقم العبارة والكلمة الأساسية (FOR) تتبع بمعادلة حسابية يكون فيها الطرف الأيسر متغير الانشوطة وفي الطرف الأيمن إما ثابت رقمي أو متغير رقمي أو تعبير حسابي، تعمل هذه المعادلة الحسابية للتمهيد للعداد الذي يمثله متغير الانشوطة. يأتي بعد المعادلة الحسابية الكلمة الأساسية (TO) تتبع بمعلمة تمثل القيمة النهائية

التي يمكن أن يصلها متغير الانشوطة. يتم تحديد مقدار الزيادة التي يمكن أن يزدادها العداد في كل تكرار للانشوطة بمعلمة تأتي بعد الكلمة الأساسية (STEP) . مما تقدم، تجد عبارة (FOR) تتكون من ثلاثة كلمات أساسية يتم بواسطتها الإشارة الى القيمة الابتدائية لمتغير الانشوطة والقيمة النهائية وأخيراً مقدار الزيادة في كل مرة تنفذ بها الانشوطة.

تكتب بعد عبارة (FOR) كل العبارات التي يجب تكرارها وتنهى تلك العبارات بعبارة (NEXT) . حيث تجدها تتركب من رقم العبارة والكلمة الأساسية (NEXT) تتبع بمتغير الانشوطة.

مثال: يجب تكرار جزء من برنامج بيسك مائة مرة. فيما يلي المخطط الانسيابي الذي يمثل ذلك الجزء. وإلى يمينه عبارات لغة بيسك التي تنقل الفكرة المنطقية وراء خوارزمية الحل. العداد هو X .



شكل (7 . 5)

ملاحظـات :

1 ـ يمكنـك حذف الكلمـة الأساسيـة (STEP) وما بعـدها في عبـارة (FOR) لأن ذلك يدل على أن مقدار الزيادة لمتغير الانشوطة هو واحد.

2 ـ يمكن أن تكـون القيمـة الابتـدائيـة لمتغيـر الانشـوطـة أكبـر من القيمـة النهائية بحيث يكون مقدار «الزيادة» هو قيمة سالبة.

```
150  FOR X=100 TO 1 STEP -1
```

3 ـ عنـد بدايـة تنفيذ الانشـوطة تقـارن القيمة الابتـدائية مـع القيمة النهـائية للتأكد من أنهـا لم تتجاوزهـا في القيمة أي أقـل أو مساويـة للقيمة النهـائية في حالة كون قيمة الزيادة موجبة. أو أكبر أو مساويـة للقيمة النهـائية عنـدما تكون قيمة «الزيادة» سالبة.

4 ـ عند تنفيذ عبارة (NEXT)، يتم زيادة قيمة متغير الانشـوطة وتقـارن قيمته مع القيمة النهائية لكي تنفذ عبارات الانشوطة في حـالة عـدم تجاوز القيمـة النهائية أو تخطي الانشوطة بأكملها إلى العبارة التي تلي عبارة (NEXT) .

5 ـ يمكن أن تكون كل من القيمة الابتدائيـة والنهائيـةومقدار الـزيادة عـدداً عشـرياً. ولكن عليـك أن تضع من ضمن حسـاباتـك التقريب النـاتـج عن التمثيل الداخلي للأعداد العشرية في الحاسبة الالكترونية.

```
150  FOR X=0.1 TO 10.8 STEP 1.5
```

مثال: سنكتب برنامج لحساب أجور السـاعات الإضـافية للمشتغلين المـائة في معمل الحاسبات الالكترونية الذي عهدته في المسـائل المحلولـة للفصـل الأول. وتذكـر أننا كتبنـا برنـامجاً بلغـة بيسك لحسـاب تلك الأجـور ولكن بـاستخـدام عبـارة (IF - THEN) وعبـارة (GOTO) ،

حيث سيطرنا على إيقاف تنفيذ الانشوطة باستخدام العداد. أما الآنفسنعيد كتابة ذلك البرنامج ولكن باستخدام عبارتي (FOR/NEXT) لتكوين والسيطرة على الانشوطة الموجودة في البرنامج.

```
10  REM AN EXAMPLE SHOWS (FOR/NEXT) LOOP
20  REM C= VARIABLE OF THE LOOP
30  REM N$ = WORKER'S NAME, H= HOURLY RATE
40  REM W= WORKED HOURS
50  REM T= OVERTIME WAGE
60  REM ***************************************************
70  REM THE LOOP IS STARTING
80  FOR C=1 TO 100 STEP 1
90  PRINT "WORKER'S NAME (N$), HOURLY RATE (H)"
100 INPUT N$ , H
110 PRINT "WORKED HOURS (W) FOR"; N$
120 INPUT W
130 LET T= H * W
140 PRINT "WORKER'S NAME", "OVERTIME WAGE"
150 PRINT "--------------", "--------------"
160 PRINT N$, T
170 NEXT C
180 STOP
190 END
```

ملاحظات :

1 ـ تجنّب تغيير قيمة متغير الأنشوطة داخل الانشوطة.

2 ـ تجنب الانتقال من خارج الانشوطة الى احدى عبارات الانشوطة الواقعة بين عبارتي (FOR/NEXT)، لأن ذلك سوف لا يؤدي الى إعطاء متغير الأنشوطة أية قيمة نتيجة لتجاوزه عبارة (FOR) عند الانتقال من الخارج الى داخل الانشوطة مباشرة.

3 ـ يجب دخول انشوطة عبارتي (FOR/NEXT) من خلال عبارة (FOR) والبقاء داخل الانشوطة الى أن يتحقق الشرط في عبارة (FOR). أو الخروج المشروط أو غير المشروط من الانشوطة والبقاء خارج الانشوطة.

4 ـ يمكن تواجد انشوطة عبارتي (FOR/NEXT) بكاملها داخل انشوطة أخرى متكون من عبارتي (FOR/NEXT) أخرى. ولكن تجنب التداخل (overlapping) بين انشوطتين. فجزء البرنامج التالي هو غير صحيح من وجهة نظر قواعد نحو لغة بيسك.

```
20   FOR A=1 TO X
30   FOR B=1 TO Y
___________
___________
___________
310  NEXT A
320  NEXT B
```

تداخل

بينما تجد جزء البرنامج التالي صحيحاً لعدم وجود التداخل أي احتواء الانشوطة بكاملها داخل انشوطة أخرى.

```
20   FOR A=1 TO X
30   FOR B=1 TO X
___________
___________
310  NEXT B
320  NEXT A
```

بحيث يتم تنفيذ الانشوطة الداخلية بالكامل في كل مرة يتم بها تنفيذ

الانشوطة الخارجية. أي سيتم تنفيذ الانشوطة الداخلية بعدد من المرات قدره (X × Y).

مما سبق نستخلص إمكانية تكوين انشوطة باستخدام عبارتي (FOR/NEXT) والسيطرة عليها أيضاً. بحيث :

(أ) تؤشر عبارة (FOR) القيمة الابتدائية والنهائية ومقدار الزيادة في قيمة متغير الانشوطة.

(ب) تستخدم عبارة (NEXT) في نهاية الانشوطة لنقل التنفيذ مرة أخرى الى عبارة (FOR).

(جـ) تعمل عبارة (FOR) على التأكد من عدم تجاوز متغير الانشوطة للقيمة النهائية. فإذا تجاوزها، نجد التنفيذ ينتقل الى خارج الانشوطة. وإذا لم يتجاوزها فتستمر الحاسبة الالكترونية في تنفيذ الانشوطة.

(د) تخضع عبارتي (FOR/NEXT) لجملة قواعد منها :

1 ـ يفترض مقدار الزيادة في القيمة الابتدائية هو واحد، إذا حذفت الكلمة الأساسية (STEP).
2 ـ يمكن إنقاص قيمة متغير الانشوطة بجعل ما يأتي بعد الكلمة الأساسية (STEP) هو قيمة سالبة.
3 ـ يمكن الانتقال الى خارج الانشوطة.
4 ـ لا تغير قيمة متغير الانشوطة أثناء تنفيذ عباراتها.
5 ـ يمكن أن تكون القيم في عبارة (FOR) إما ثوابت أو متغيرات أو تعابير حسابية.

6 ـ إذا تطلب منطق حـل مسألـة أن يكون هنـالك أكثـر من انشوطـة. فيجب أن لا تتـداخـل تلك الانشـوطـات، بـل يجب أن تتـواجـد الانشوطة بالكامل ضمن انشوطة أكبر أو تنفصل عنها كلياً.

7 ـ لا يجب الانتقـال مباشـرة الى احدى عبـارات الانشوطـة الـواقعـة داخــل عبـارتي (FOR/NEXT) بـدون المـرور من خــلال عبـارة (FOR) عند التنفيذ

اختبر نفسك:

1 - هل يمكنك الاستعاضة عن مجموعة عبارات (IF - THEN) التالية باستخدام عبارة واحدة فقط من لغة بيسك كيف؟

```
40  IF X = 1 THEN 10
50  IF X = 2 THEN 170
60  IF X = 3 THEN 30
70  IF X = 4 THEN 210
80  IF X = 5 THEN 300
```

2 - هل سيتم تنفيذ الأنشوطة التالية؟ اذكر السبب

```
10  FOR A = 1 TO 99
20  A = 1 + 1
30  X = X ↑ 3
40  PRINT "X ="; X
50  NEXT A
```

3 - كم مرة سيتم تنفيذ الأنشوطة الداخلية في جزء البرنامج التالي:

```
10  FOR A = 1 TO 3
20  FOR B = 1 TO 3
30  PRINT "X ="; X
40  NEXT B
50  NEXT A
```

4 - ما الفرق من حيث الوظيفة بين عبارات الانتقال التالية في لغة بيسك :

IF .... THEN ....

GOTO

ON ..... GOTO

5 - ما هي أهم القواعد التي يجب مراعاتها عند تكوين أنشوطة باستخدام عبارتي (FOR .NEXT) .

6. المصفوفات ، احدى تطبيقات الشوطة عبارتي(FOR/NEXT) :

ذكرنا في الفصل الثاني أن بإمكان لغة بيسك معالجة ثلاثة أنواع من المتغيرات هي المتغيرات الرقمية والوصفية والمتغيرة ذات الأبعاد. يمكن أن يكون المتغير ذو بعد واحد. أو بعدين ويمكن أن يمثل المتغير ذو الأبعاد ثوابت رقمية أو ثوابت مقطعية. راجع فقرة رقم (5) من الفصل الثاني .

مثال : لاحظ المتغير ذو البعد الواحد فيما يلي :

X(1) = 10.3
X(2) = 5.9
X(3) = 2.8
X(4) = 6.1

يمكن إدخال هذا المتغير بواسطة عبارة (LET) كما يلي :

```
10  LET X(1) = 10.3
20  LET X(2) = 5.9
30  LET X(3) = 2.8
40  LET X(4) = 6.1
```

ولكن ما العمل إذا كانت عدد البيانات التي يمثلها المتغير ذو الأبعاد كبير. يمكننا إدخال بيانات المتغير ذو الأبعاد بواسطة عبارتي (FOR/NEXT) كما يلي :

```
40   FOR I=1 TO 4
50   READ X(I)
60   NEXT I
70   DATA 10.3, 5.9, 2.8, 6.1
```

ولكن يجب حجز مواقع في ذاكرة الحاسبة الالكترونية مسبقاً للبيانات التي يمثلها المتغير ذو الأبعاد. يتم ذلك بواسطة عبارة (DIM)، وهي مختصر كلمة (DIMENSION) أي (بُعد) وجمعها (أبعاد).

```
20   DIM X(4)
40   FOR I=1 TO 4
50   READ X(I)
60   NEXT I
70   DATA 10.3, 5.9, 2.8, 6.1
```

مثال: اخزن بيانات على شكل جدول فيه أربعة صفوف وثلاثة أعمدة، وكمامبين في الجدول أدناه:

1	2	3
4	5	6
7	8	9
10	11	12

الحل: سنستخدم متغير ذو بعدين Y(4 , 3)، بحيث يتم خزن البيانات في مواقع ذاكرة الحاسبة الالكترونية بواسطة برنامج البيسك التالي:

```
10 DIM Y(4 , 3)
20 FOR I=1 TO 4
30 FOR J=1 TO 3
40 READ Y(I , J)
50 NEXT J
60 NEXT I
70 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
```

من الأمثلة اعلاه وجدت أن عبارة (DIM) تتركب من رقم العبارة والكلمة الأساسية (DIM) تتبع بمتغير ذو أبعاد أو أكثر يفصل بينهم فارزة.

مثال : اخزن معلومات عن خمسة من اصدقائك، تتضمن اسم الصديق، يوم مولده، رقم تلفونه، طوله بالأمتار. ثم اطبع تلك المعلومات عند كل مرة ينفذ بها ذلك البرنامج. اجعل البرنامج مرناً بحيث يمكنك زيادة عدد اصدقائك وقت الرخاء أو إنقاصهم وقت الشدة.

الحل : سنضع أسماء الأصدقاء مع المعلومات المتعلقة بكل منهم في جدول، كما يلي :

طوله بالأمتار HEIGHT	رقم تلفونه PHONE NO.	يوم مولده BIRTH DAY	اسم الصديق FRIEND'S NAME
1.7	5541401	1960/9/10	مصطفى
1.9	7768213	1956/2/19	رياض
1.8	4417952	1959/5/28	أسامة
1.75	7729892	1955/7/20	زيد
2.0	4434410	1958/9/3	كامل

يمكن خزن أسماء الأصدقاء مع يوم مولد كل منهم تحت اسم متغير ذو بعدين A$(5 , 2) والذي يمكن تصوره كما يلي :

	, 1)	, 2)
A$(1	MOSTAFA	10/9/1960
A$(2	RIAD	19/2/1956
A$(3	USAMA	28/5/1959
A$(4	ZAID	20/7/1955
A$(5	KAMEL	3/9/1958

وسنخزن رقم تلفون وطول الأصدقاء تحت اسم متغير ذو بعدين آخرين هو B(5 , 2) .

	,1)	,2)
B (1	5541401	1.7
B (2	7768213	1.9
B (3	4417952	1.8
B (4	7729892	1.75
B (5	4434410	2.0

عند كتابة برنامج بيسك يتعامل مع متغيرات ذات أبعاد، يجب حجز مواقع في الذاكرة لبيانات المتغيرات ذات الأبعاد.

```
10  DIM A$(5 , 2), B (5 , 2)
```

ولجعل البرنامج أكثر مرونة في تقبل الزيادة أو النقصان لعدد الأصدقاء أو المعلومات المتعلقة بهم، فسنجعل أبعاد المتغيرات متغيرة وليست ثوابت رقمية كما يلي :

```
10  LET X=5
20  LET Y=2
30  DIM A$(X , Y), B (X , Y)
```

الآن سندخل بيانات المتغيرين ذوي الأبعاد، ومن ثم نطبعهما تحت عناوين مناسبة وكما يلي من عبارات لغة بيسك.

```
40  PRINT "A FRIEND'S NAME", "HIS PHONE NO.",
50  PRINT "HIS BIRTH DAY", "HIS HEIGHT"
60  PRINT "---------------------------------", "---------------------------------",
70  PRINT " ---------------------------------- ", " -----------------------------------"
80  FOR I=1 TO X
90  FOR J=1 TO Y
100 READ A$(I , J)
110 NEXT J
120 FOR K=1 TO Y
130 READ B (I , K)
140 NEXT K
150 FOR L=1 TO Y
160 PRINT A$(I , L), B (I , L)
170 NEXT L
180 NEXT I
190 DATA "MOSTAFA", "10/9/1960", 5541401, 1.7
200 DATA "RIAD", "19/2/1956", 7768213, 1.9
210 DATA "USAMA", "28/5/1959", 4417952, 1.8
220 DATA "ZAID", "20/7/1955", 7729892, 1.75
230 DATA "KAMEL", "3/9/1958", 4434410, 2.0
240 STOP
250 END
```

غذي البرنامج أعلاه إلى حاسبتك الالكترونية ونفذ البرنامج باستخدام أوامر لغة بيسك التي عهدتها في الفصل الرابع من هذا الكتاب. وتحقق من أن النتيجة التي ستظهر لك هي ذاتها التي وضع البرنامج من أجل انجازه . كما حاول وضع خوارزمية حل إما بشبه الشفرة أو بمخطط انسيابي مناسب.

7. وضع برنامج صغير للعمل داخل برنامج كبير :

عندما تكون المسألة المطلوب برمجتها معقدة، فيستحسن استخدام أسلوب البرمجة الهيكلية على تجزئة المسألة إلى مهمات صغيرة يتم برمجتها على شكل برامج فرعية (dubrotine) يتم استدعائها من قبل البرنامج الرئيسي بواسطة عبارة (GOSUB)، وبعد تنفيذ البرنامج الفرعي تعود الحاسبة الالكترونية لتنفيذ العبارة التي تلي عبارة (GOSUB).

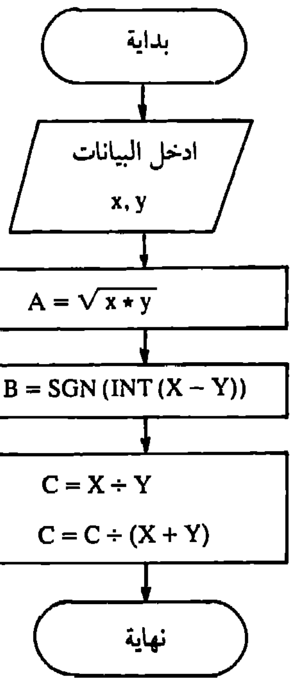
تتركب عبارة (GOSUB) من رقم العبارة والكلمة الأساسية (GOSUB) تتبع برقم أول عبارة من البرنامج الفرعي. أما عبارة (RETURN) فتكون آخر عبارة في البرنامج الفرعي. وتتكون من رقم العبارة والكلمة الأساسية (RETURN) فقط.

مثـال :

```
30   GOSUB 70
40   --------------
50   ------------------
60   -------------------
70   REM SUBROTINE IS STARTING HERE
80   -------------
90   ---------------------
100  RETURN
```

تم شـرح فكرة البـرمجة الهيكليـة في الفقـرة (7) من الفصـل الأول. وقد تم حل مسألة رقم (5) من المسائل المحلولة كمثال عليها.

مثـال: سنجري مجمـوعة عمليـات رياضيـة على بيانـات مدخلة بـاستخدام اسلوب البـرمجة الهيكليـة، وكما مـوضـح في المخـطط الانسيبـاي أدناه:



شكل (8 . 5)

سنجعل كل معالجة للبيانات المدخلة أعلاه هي برنامج فرعي:

```
10  REM STRUCTURED PROGRAMMING
20  INPUT X , Y
30  GOSUB 80
```

```
40  GOSUB 130
50  GOSUB 180
60  STOP
70  END
80  REM *****************************************************
90  REM A SUBROTINE
100 LET A= SQR (X * Y)
110 PRINT "THE SQUARE ROOT OF X * Y ="; A
120 RETURN
130 REM *****************************************************
140 REM A SUBROUTINE
150 LET B= SGN (INT (X+Y))
160 PRINT "SGN (INT (X+Y)) ="; B
170 RETURN
180 REM *****************************************************
190 REM A SUBROUTINE
200 LET C= X/Y
210 LET C=C/ (X+Y)
220 PRINT "C ="; C
230 RETURN
```

نفّذ البرنامج اعلاه على حاسبتك الالكترونية وتأكد من نتيجته.

ملاحظات :

1 ـ يمكنك وضع البرنامج الفرعي أينما شئت.

2 ـ يمكن استخدام البرنامج الفرعي في البرنامج الرئيسي بذكر رقم أو عبارة منه بواسطة عبارة (GOSUB) .

3 ـ يمكن إعادة استخدام البرنامج الفرعي عدة مرات.

4 ـ يمكنك اكتشاف وتصحيح أخطاء البرنامج بسهولة نتيجة لتجزئة مهمات البرنامج الى برامج فرعية.

5 ـ يمكنك الآن توثيق البرنامج بسهولة عن طريق توثيق برامجه الفرعية.

اختبر نفسك :

1 ـ اكتب جزء من برنامج لقراءة أسماء مائة عامل مع رواتبهم الشهرية باستخدام المصفوفات.

2 ـ اكتب صيغة العبارات التالية من خلال مثال لكل منهما.

(أ) عبارة (DIM) .

(ب) عبارة (GOSUB) .

(جـ) عبارة (RETURN) .

3 ـ اكتشف الخطأ في العبارات التالية:

```
10  DIM A$(100), B(X), V 20

50  GOSUB

100  RETURN TO 310
```

4 ـ ما الخطأ في جزء البرنامج التالي:

```
10  DIM X(20)
20  FOR I=1 TO 99
30  INPUT X(I)
40  NEXT I
```

5 ـ اكتب جميع عبارات لغة بيسك التي تتواجد كلمة (GO) أي (اذهب) من ضمن كلماتها الأساسية؟

8. مسائل برمجة محلولة :

مسألة (1) : هذه المسألة مقتبسة من سؤال ورد في ورقة امتحان (Welsh Joint Education Committee) .

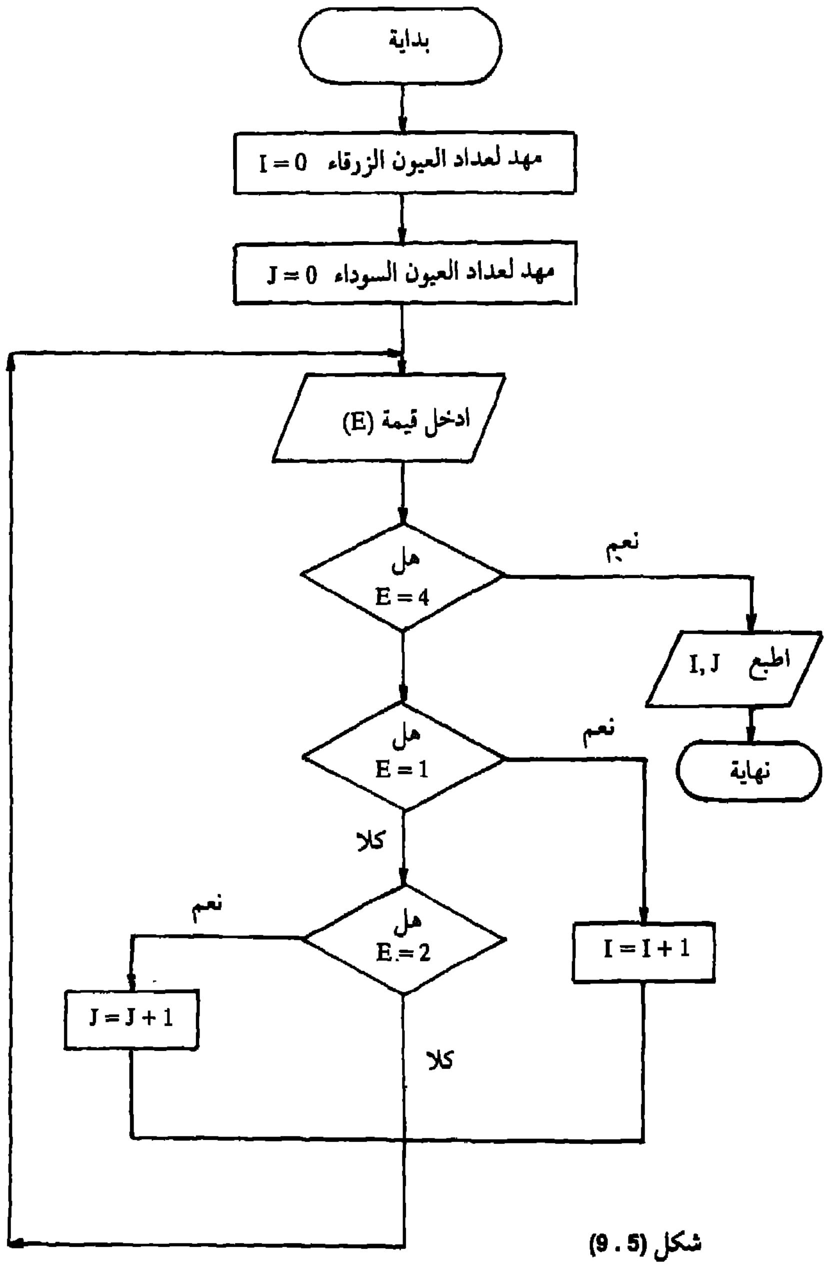
في تجربة لمعرفة عدد الطلاب ذوي العيون الزرق أو السود، تم اختبار عينة عشوائية متكونة من مائة طالب وقد ملأت الاستمارات بالمعلومات (E) كما يلي :

العين زرقاء ← E=1

العين سوداء ← E=2

للعين لون آخر ← E=3

وقد تم إعلان نهاية المعلومات بقيمة ملحقة أو ذيلية مقدارها (E=4) . برمج باستخدام لغة بيسك المخطط الانسيابي التالي ، لمعرفة عدد الطلاب من ذوي العيون الزرقاء أو السوداء.



بداية
مهد لعداد العيون الزرقاء I = 0
مهد لعداد العيون السوداء J = 0
ادخل قيمة (E)
هل E = 4
نعم
اطبع I, J
نهاية
هل E = 1
نعم
كلا
I = I + 1
هل E = 2
نعم
J = J + 1
كلا

شكل (9 . 5)

الحل: بسبب وجود خوارزمية الحل على شكل مخطط انسيابي فسنكتب البرنامج منه مباشرة :

```
10  REM A BIOLOGY EXPERIMENT
20  REM I= THE COUNTER OF BLUE EYES
30  REM J= THE COUNTER OF BLACK EYES
40  LET I=0
50  LET J=0
60  PRINT "TYPE THE CODE OF YOUR EYE'S COULOR"
70  INPUT E
80  ON E GO TO 90, 100, 60, 130
90  LET I= I+1
100 GO TO 60
110 LET J= J+1
120 GO TO 60
130 PRINT "THE NUMBER OF BLUE EYES="; I
140 PRINT "THE NUMBER OF BLAK EYES="; J
150 STOP
160 END
85  GO TO 60
```

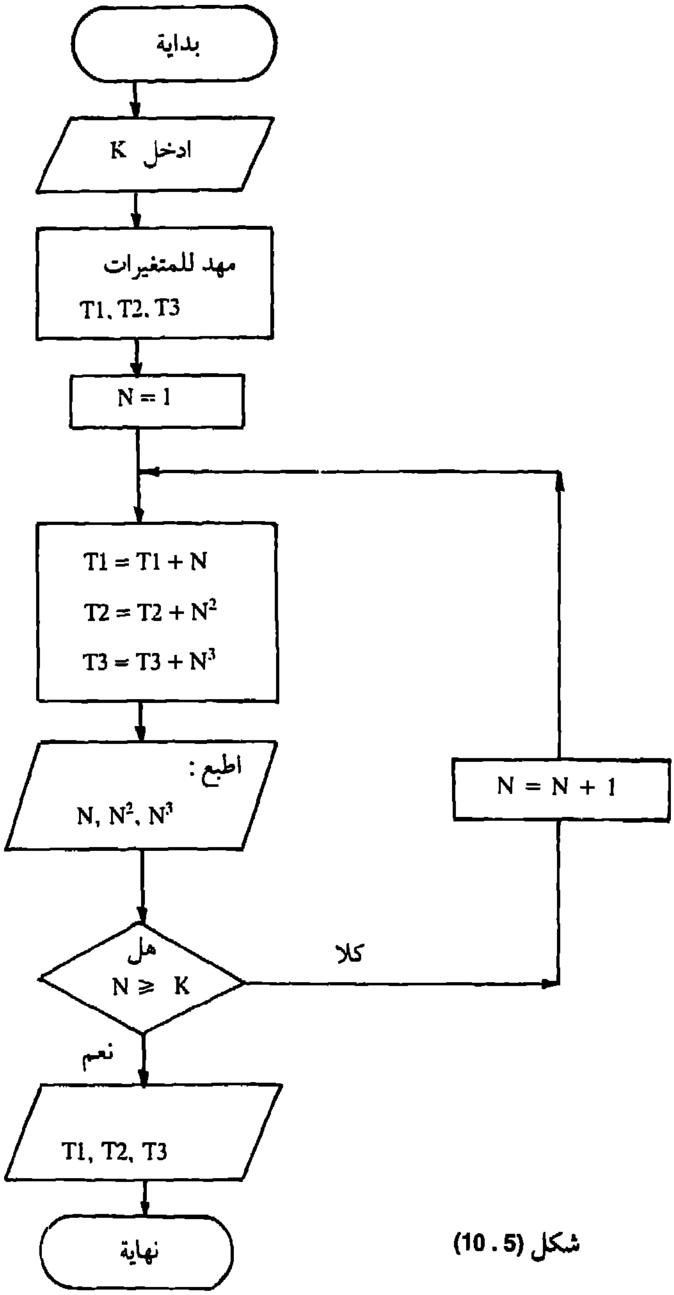
ملاحظات :

1 ـ لاحظ ما اختصرته عبارة (ON/GOTO) من عبارات لو استخدمنا بدلها عبارات (IF - THEN) .

2 ـ استخدمت عبارة (GOTO) لتكوين انشوطة .

3 ـ استخدمت عبارة (ON/GOTO) لنقل خطوات ثلاثة من رموز اتخاذ القرار في المخطط الانسيابي .

4 ـ استخدمت القيمة الملحقة أو الذيلية للسيطرة على عدد المرات التي تنفذ بها الانشوطة .


بداية
ادخل K
مهد للمتغيرات
T1, T2, T3
N = 1
T1 = T1 + N
T2 = T2 + N2
T3 = T3 + N3
اطبع :
N, N2, N3
N = N + 1
هل
N ≥ K
كلا
نعم
T1, T2, T3
نهاية

شكل (5 . 10)

مسألة (2) : هـذه المسألـة مقتبسـة من سؤال ورد في ورقـة امتحـان (University of Cambridge, local Examination Syndicates) . اشـرح هـدف المخطط الانسيـابي التـالي، واكتب بـرنـامـج بيسـك يعبـر عن خوارزمية الحل في المخطط الانسيابي التالي :

الحــل : إن هدف البرنامج هو:

أولاً : طبع قيمة كل خطوة من عداد (N) مع مربعها ومكعبها .

ثانياً : طبع المجموع الكلي لقيم خـطوات العداد (N) مـع المجموع الكلي لمربعها والمجموع الكلي لمكعبها.

أما برنـامج البيسـك فستكون كتـابته مبـاشرة من المخـطط الانسيابي وكما يلي :

```
10  REM AN EXAMPLE TO FIND TOTALS
20  REM N= COUNTER = K
30  REM T1= TOTAL SUM OF THE (N'S)
40  REM T2= TOTAL SUM OF THE (N)²
50  REM T3= TOTAL SUM OF THE (N)³
60  PRINT "TYPE THE COUNTER'S MAXIMUM VALUE"
70  INPUT K
80  LET T1=0
90  LET T2=0
100 LET T3=0
110 FOR N=1 TO K
120 LET T1= T1+N
130 LET T2= T2+ N ↑ 2
140 LET T3= T3+ N ↑ 3
150 PRINT "N=": N, "N²="; N ↑ 2, "N³="; N ↑ 3.
160 NEXT N
170 PRINT "T1="; T1, "T2="; T2, "T3="; T3
```

```
180 STOP
190 END
```

ملاحظـات :

1 ـ استخدمنا عبـارتي (FOR/NEXT) لتكوين والسيطرة على انشوطة.

2 ـ يمكن برمجة المخطط الانسيابي بـاستخدام عبـارة (IF - THEN) لاختبار شرط الخروج من الانشوطة والعبارة (GOTO) لتكوين انشوطة.

مسـألة (3) : هـذه المسألـة مقتبسة من سؤال ورد في ورقـة امتحان (Welsh Joint Education Committee).

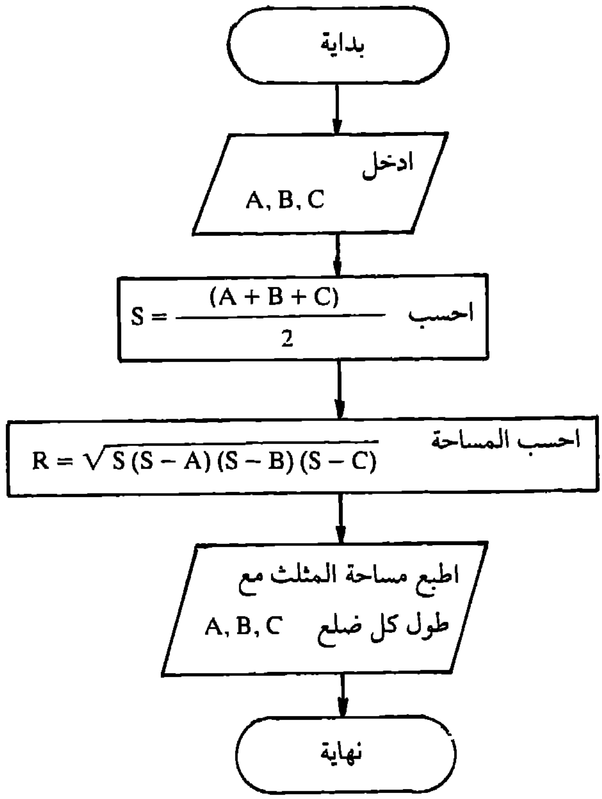
يمثـل المخطط الانسيـابي أدناه خـوارزميـة إيجـاد مسـاحـة مثلث باستخدام صيغة هيرون (Heron) :

$$R = \sqrt{S(S-A)(S-B)(S-C)}$$

حيث أن

$$S = \frac{A+B+C}{2}$$

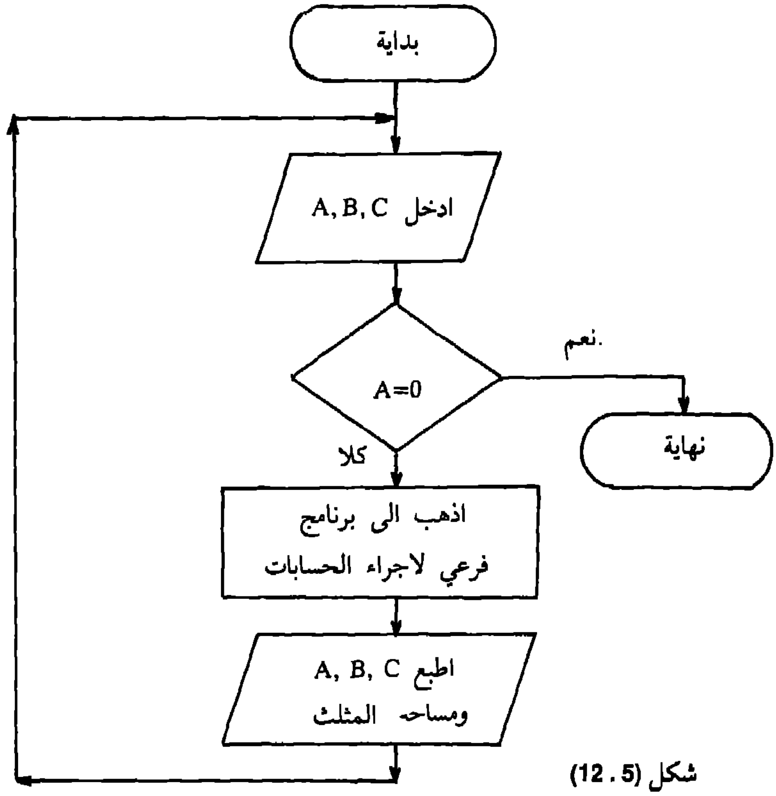
اكتب برنامـج بلغة بيسـك يمكن بواسـطته حسـاب المساحـة لعدد غيـر محدد من المثلثات. يمكنك تحوير المخطط الانسيـابي ليناسب هـدف البرنامج.



الحــل : يمكننـا جعل البـرنامـج يحسب المسـاحـة لعـدد غيـر محـدد من المثلثـات وذلك بـإدخال البيـانات بـواسطة عبـارة (INPUT) . ويمكننـا إيقاف تنفيذ البرنامـج بإدخـال قيمة احـدى المتغيرات مسـاوية للصفـر. وليكن المتغير (A=0) . ويتم ذلك بواسطة عبارة (IF - THEN) .

نستـطيع جعـل الحسابـات المطلوب إجـرائها في بـرنامـج فـرعي منفصل عن البرنامج الرئيسي .

يمكن تحوير المخـطط الانسيابي لكي ينجـز هدف المسـألة كمـا يلي :



شكل (12 . 5)

```
10  REM COMPUTE THE AREA OF TRIANGLES
20  REM A, B, C ARE THERE SIDES OF A TRIANGLE
30  LET S=0
40  LET R=0
50  PRINT "TYPE THE VALUES OF THE"
60  PRINT "THREE SIDES OF TRIANGLE"
70  INPUT A, B, C
80  IF A=0 THEN 130
90  GOSUB 150
100 PRINT    "A       B       C       TRIANGLE       AREA"
110 PRINT "---------------------------------------------------------------"
120 PRINT A, B, C, R
130 STOP
```

```
140  END
150  REM  **************************************************
160  REM SUBROUTINE TO COMPUTE THE AREA
170  LET S= (A+B+C)/2
180  LET R= SQR (S * (S-A) * (S-B) * (S-C))
190  RETURN
200  REM  **************************************************
```

نفـذ البرنـامج اعـلاه وتأكـد من أن نتائجـه كما تم التخـطيط لهـا في المخطط الانسيابي كخوارزمية للحل .

9 خلاصة:

كرست مفردات هذا الفصل لتفسير عبارات السيطرة في لغة بيسك. حيث تجد في هذه اللغة من العبارات التي تستخدم لتكوين انشوطة والسيطرة عليها مما يحقق الفائدة من الحاسبة الالكترونية في إمكانياتها على تكرار الحسابات. يمكن تلخيص ما جاء في هذا الفصل بالنقاط التالية:

أولاً : يمكن تكوين انشوطة بواسطة عبارة (GOTO) ويسيطر عليها بواسطة عبارة (IF - THEN) .

ثانياً : يمكن السيطرة على تكرار تنفيذ انشوطة في لغة بيسك بثلاثة أساليب منها:

(أ) باستخدام العداد.

(ب) باستخدام القيمة الملحقة أو الذيلية.

(جـ) باستخدام عبارتي (FOR/NEXT) .

ثالثاً : هنالك خمسة أنواع من الانتقال لتنفيذ عبارات برنامج بيسك:

(أ) الانتقال غير المشروط باستخدام عبارة (GOTO) .

(ب) الانتقال المشروط باستخدام عبارة (IF - THEN) .
(جـ) الانتقال الانتقائي باستخدام عبارة (ON/GOTO) .
(د) الانتقال الى برنامج فرعي باستخدام عبارة (GOSUB) .
(هـ) الانتقــال الى بــرنــامــج رئيسي بــاستخــدام عــبــارة (RETURN) .

رابعاً : في المخطط الانسيابي تمثل العبارات المـذكورة أدنـاه من لغة بيسك كما هو مؤشر ازاء كل منها :
(أ) تمثل عبارة (GOTO) بخط انتقال يربط بين صندوقين .
(ب) تمثل عبارة (IF - THEN) برمز اتخاذ القرار المعيني .
(جـ) تمثل عبارة (ON/GOTO) بعدة رموز اتخاذ قرار معينية .
(د) تمثل عبارتي (FOR/NEXT) برمز القرار المعيني .

خامساً : يمكننا الحجز المسبق لبيانات متغير ذو أبعاد بـاستخدام عبـارة (DIM) .

سادساً : في المسـائل المعقـدة، تستخـدم البـرمجـة الهيكليـة لتجـزئـة البرنامج الى مهمات صغيرة، والتي تسمى بالبـرامج الفـرعية. بحيث يتم الانتقال الى تنفيذ تلك البـرامج من بـرنامـج رئيسي بـواسـطة عبـارة (GOSUB) وتتم العـودة الى العبـارة التي تلي عبارة (GOSUB) بواسطة عبارة (RETURN) .

سابعاً : يمكننـا تلخيص ما طـرح بخصوص العبـارة (ON/GOTO) بما يلي :
(أ) يعتبر هذا الانتقال انتقائي شرطي .
(ب) إذا كانت نتيجة التعبير الحسابي عدد عشري فـإنها تـأخذ

العدد الصحيح فقط من العدد الكلي.

(جـ) لا تنفذ عبارة (ON/GOTO) إذا كانت نتيجة التعبير الحسابي أو قيمة المتغير الرقمي صفر أو عدداً صحيحاً أكبر من عدد أرقام العبارات المتوفرة.

(أ) من تطبيقات هذه العبارة هو اختيار معلومات المستفيد باختبار (اختار الجواب الصحيح). أو بتوفير قائمة من الاختيارات المقترح إجرائها من قبل الحاسبة الالكترونية لصالح الجهة المستفيدة.

ثامناً : تخضع عبارتي (FOR/NEXT) لجملة قواعد منها:

(أ) يفترض مقدار الزيادة في القيمة الابتدائية هو واحد إذا حذفت الكلمة الأساسية (STEP) .

(ب) يمكن انقاص قيمة متغير الانشوطة بجعل ما يأتي بعد الكلمة الأساسية (STEP) هو قيمة سالبة.

(جـ) لا يمكن الانتقال الى داخل الانشوطة مباشرة، بل يجب المرور من خلال عبارة (FOR) أولاً.

(د) لا تغير قيمة متغير الانشوطة أثناء تنفيذ عبارات الانشوطة.

(هـ) يمكن أن تكون القيم في عبارة (FOR) إما ثوابت أو متغيرات أو تعابير حسابية.

(و) إذا تطلب منطق حل مسألة أن يكون هنالك أكثر من انشوطة، فيجب أن لا تتداخل تلك الانشوطات، بل يجب أن تتواجد الانشوطة بالكامل ضمن انشوطة أكبر، أو تنفصل بالكامل عنها.

10. كلمات قد تعلمتها :

عبارة «اذهب الى»	GO TO
عبارة «إذا كان ـ فإن»	IF - THEN
العداد	counter
قيمة ملحقة أو ذيلية	Trailer Value
عبارة الانتقال الانتقائي	ON / GOTO
عبارة للسيطرة على انشوطة	FOR... TO... STEP
عبارة تكوين انشوطة	NEXT
المصفوفات	ARRAY
عبارة بعد (جمعها أبعاد)	DIM
البرمجة الهيكلية	structured programming
عبارة اذهب الى برنامج فرعي	GOSUB
عبارة ارجع أو عودة الى برنامج رئيسي	RETURN
برنامج فرعي	subroutine

11. تمارين (5) :

استخدم كلمات قد تعلمتها.

(أ) متغير الانشوطة	(و) قيمة ملحقة أو ذيلية
(ب) (FOR)	(ز) البرنامج الفرعي
(جـ) للمصفوفة	(ح) المشروط
(د) (IF - THEN)	(ط) الانشوطة
(هـ) (ON/GOTO)	(ى) (GOTO)

1 ـ تستخدم عبارة ـــــــــ لنقل تنفيذ عبارات البرنامج بانتقاء عبارة واحدة من مجموعة عبارات.

2 ـ بواسطة اختبار تعبير متراجحة يتم اتخاذ قرار الانتقال في عبارة ـــــــــ .

3 ـ يحسب ـــــــــ عدد المرات التي تنفذ بها الانشوطة.

4 ـ يمكن تكوين انشوطة بواسطة عبارة ـــــــــ .

5 ـ يسمى التنفيذ المتكرر لسلسلة من العبارات بـ ـــــــــ .

6 ـ تقارن عبارة ــــــــــ قيمة متغير الانشوطة مع القيمة النهائية لعدد المرات التي يجب أن تنفذ بها الانشوطة .

7 ـ يتم الانتقال ــــــــــ بواسطة عبارة (IF - THEN) .

8 ـ تستخدم عبارة (DIM) لخزن مواقع في الذاكرة مسبقاً ــــــــــ .

9 ـ يتم الانتقال الى ــــــــــ بواسطة عبارة (GOSUB) .

10 ـ توضع ــــــــــ في نهايةِ قائمة بيانات للسيطرة على تنفيذ البرنامج .

صح أم خطأ :

1 ـ بعد تنفيذ برنامج فرعي يتم الرجوع الى البرنامج الرئيسي بواسطة عبارة (GOTO) .

2 ـ تستخدم المصفوفة للسيطرة على تنفيذ انشوطة .

3 ـ في المخطط الانسيابي ، تمثل عبارة (ON/GOTO) بأكثر من رمز قرار واحد .

4 ـ تسبب عبارة (GOTO) الانتقال الانتقائي إلى أحد عبارات مجموعة من عبارات برنامج بيسك .

5 ـ يوضع العداد في نهاية قائمة بيانات للسيطرة على تنفيذ انشوطة .

6 ـ للعداد والقيمة الملحقة الوظيفة نفسها في السيطرة على تنفيذ انشوطة .

7 ـ يتم اقتطاع الجزء العشري لقيمة التعبير الحسابي في عبارة (ON/GOTO) واستخدام الجزء الصحيح فقط.

8 ـ يجب استخدام كل من عبارة (FOR) وعبارة (NEXT) معاً في برنامج بيسك لتكوين انشوطة والسيطرة عليها.

9 ـ يمكن أن يكون متغير الانشوطة إما متغير رقمي أو متغير وصفي.

10 ـ يجب أن تذكر قيمة الزيادة بعد الكلمة الأساسية (STEP) كلما ذكرت عبارة (FOR) .

أجب باختصار :

1 ـ ما هي الأساليب في لغة بيسك التي يمكن بواسطتها السيطرة على تنفيذ انشوطة؟

2 ـ كم نوع من الانتقال يمكن أن يحدث في برنامج بيسك؟ اذكر تلك الأنواع.

3 ـ كيف تمثل العبارات التالية في لغة بيسك بمخطط انسيابي معين:

(أ) عبارة (FOR/NEXT) .

(ب) عبارة (GOTO) .

(جـ) عبارة (ON/GOTO) .

4 ـ استعض عن العبارات الأربعة التالية بعبارة واحدة فقط.

```
 90  IF A=1 THEN 200
100  IF A=2 THEN 150
110  IF A=3 THEN 330
120  IF A=4 THEN 20
```

5 ـ اكتب جزء من برنامج تكون بدايته كما يلي :

```
10   DIM X (10, 15)
```

بحيث يتم قراءة البيانات من مصفوفة ذات بعدين.

6 ـ اكتب صيغة العبارات التالية من خلال مثال :

(أ) (IF - THEN) .

(ب) (GOSUB) .

(جـ) (RETURN) .

7 ـ ما الخطأ في جزء البرنامج التالي :

```
50   FOR I=1 TO X
60   I= I+1
70   PRINT "I="; I
80   NEXT X
```

8 ـ هل سيتم الانتقال الى عبارة رقم (190) ؟

```
30   IF (X * Y) ↑ 2 > 55 THEN 190
```

إذا كانت

X=5 ، Y=1

9 ـ كم مرة سيتم تنفيذ الانشوطة الداخلية في جزء برنامج البيسك التالي :

```
30   FOR I=1 TO 9
40   FOR J=1 TO 5
50   PRINT "X="; X
60   NEXT I
70   NEXT J
```

10 ـ كيف يمكن تجسيد فكرة البرمجة الهيكلية في لغة بيسك؟ اذكر العبارات مع الأمثلة المناسبة.

اختار الجواب الصحيح :

1 ـ كم قيمة الزيادة التي سيأخذها متغير الانشوطة في العبارة التالية :

```
30 FOR I=9 TO 1
```

(أ) (1−) .

(ب) (2) .

(جـ) (3) .

(د) العبارة ناقصة .

2 ـ اختار العبارة الصحيحة مما يلي :

(أ) `10 FOR I=1 TO 9 STEP 11`

(ب) `20 FOR K=99 TO 98 STEP −5`

(جـ) `30 FOR L=1 TO 5`

(د) `40 FOR M=1 TO 92 RETURN`

3 ـ ماذا ستكون قيمة المتغير (X) بعد الانتهاء من تنفيذ البرنامج التالي :

```
10 FOR I=1 TO 3
20 READ A
30 LET X= I+A
40 NEXT I
50 DATA 13, 41, 30, 90
60 STOP
70 END
```

(أ) (14) .

(ب) (33) .

(جـ) (42) .

(د) (97) .

4 ـ أي من العبارات التالية ستسبب انتقال غير مشروط؟

(أ) 40 IF A=89 THEN 20

(ب) 120 ON B GOTO 210, 101, 310, 191

(جـ) 70 GOTO 230

(د) 20 GOTO A$

5 ـ أي من العبارات التالية غير صحيحة من وجهة نظر قواعد نحو لغة بيسك؟

(أ) 40 IF A≠1 THEN 350

(ب) 40 IF A$ = "SAMI" THEN 350

(جـ) 40 IF X> = 2 THEN 350

(د) كل من (ب) و(جـ) .

6 ـ إذا اعطيت عبارتي بيسك التاليتين:

```
99  LET A=100
101 ON A/30 GO TO 20, 40, 60, 80
```

فسينتقل تنفيذ العبارات إلى:

(أ) عبارة رقم (40) .

(ب) عبارة رقم (20) .

(جـ) عبارة رقم (60) .

(د) عبارة رقم (80) .

7 ـ يعبر عن العبارة (ON/GOTO) في المخطط الانسيابي:

(أ) برمز إدخال البيانات المتوازي الأضلاع .

(ب) برمز معالجة البيانات المستطيل .

(جـ) بخط انسيابي .

(د) بعدة رموز لاتخاذ القرار.

8 ـ تخضع عبارة (FOR) لواحد فقط من الأمور التالية :

(أ) لا يكون مقدار للزيادة إذا حذفت الكلمة الأساسية (STEP) .

(ب) لا يمكن جعل القيمة سالبة بعد الكلمة الأساسية (STEP) .

(جـ) يمكن أن تكـون القيم في عبارة (FOR) إمـا ثوابت أو متغيـرات أو تعابير حسابية .

(د) لا يتوجب ارتباط عبارة (NEXT) معها في برنامج بيسك .

9 ـ يكون الانتقال انتقائياً في :

(أ) عبارة (GOTO) .

(ب) عبارة (IF - THEN) .

(جـ) عبارة (ON/GOTO) .

(د) عبارة (GOSUB) .

10 ـ إن وجود برنامج رئيسي متبـوع بعد بـرامج فـرعية يمكن تنفيـذه بمجرد ذكـر رقـم أول عبـارة في عبـــارة (GOSUB) والعـودة الى البـــرنـامــج الرئيسي بعبارة (RETURN) . يسمى هذا الأسلوب ـــــــــ .

(أ) بالانتقال المشروط.

(ب) بالانتقال غير المشروط.

(جـ) بالبرمجة الهيكلية .

(د) بالمصفوفة .

الملاحق

ملحق | أ | :

مصادر عربية مقترحة للدراسة وزيادة الاطلاع

في لغة بيسك

1 ـ محمد السعيد خشبه: «أساليب تخطيط البرامج بلغة بيسك»، (1985) .

2 ـ عبد العزيز فهمي هيكل وج. مدبك: «التطوير الالكتروني للمكاتب»، دار الراتب الجامعية، (1985) .

3 ـ ع. مكداشي وج. مدبك: «استخدام الكومبيوتر في الأرشيف»، دار الراتب الجامعية، (1985) .

4 ـ عبد العزيز فهمي هيكل: «الكمبيوتر وإدارة الأعمال (1)»، دار الراتب الجامعية ، (1985) .

5 ـ عبد العزيز فهمي هيكل: «الكمبيوتر وإدارة الأعمال (2)»، دار الراتب الجامعية، (1985) .

6 ـ عبد العزيز فهمي هيكل: «الكمبيوتر والاقتصاد القياسي»، دار الراتب الجامعية، (1985) .

7 ـ ج. مدبك: «استخدام الكمبيوتر في المختبرات والعيادات الطبية والمستشفيات»، دار الراتب الجامعية، (1985) .

8 ـ مظهر طايل: «موسوعة الكمبيوتر لغة وأداء (1)»، دار الراتب الجامعية، (1986) .

9 ـ عبد الحكيم عبد الله: «الميكروكمبيوتر والبرمجة للمبتدئين بلغة بيسك»، دار الراتب الجامعية، (1986) .

10 ـ مجلس البحث العلمي العراقي ـ مركز البحوث الالكترونية والحاسبات: «أساسيات برمجة الحاسبات الشخصية»، (1986) .

11 ـ ليلى صالح العلي: «الحاسبات الالكترونية الشخصية»، دار واسط، (1986) .

12 ـ سامي الرماحي: «مقدمة تعريفية عن الحاسبات الالكترونية»، دار الراتب الجامعية (1987) .

13 ـ سامي الرماحي: «كيف تفكر عند برمجة حاسبة الكترونية»، دار الراتب الجامعية (1987) .

14 ـ سامي الرماحي وعدنان الحمداني: «محو الأمية في مجال الحاسبات الالكترونية»، كتاب مترجم، دار آفاق عربية للنشر، (1988) .

ملحق ب :

مصادر انكليزية مقترحة للدراسة وزيادة الاطلاع حول لغة بيسك

1 J. G. Kemeny and T. E. KURTZ:"**BASIC programming**", JOHN WILEY and SONS, INC., Second edition, (1971).

2. D. ALCOCK: "**ILLUSTRATING BASIC**", CAMBRIDGE UNIVERSITY PRESS, (1978).

3. D. M. Monro: **basic BASIC**", Edward Arnold, (1979).

4. P. Bishop: "**Computer Programming in BASIC**",.

5. R. Hunt: "**Pocket Guide to BASIC**", PITMAN BOOKS LTD, (1983).

6. B. S. , GOTTFRIED:"**PROGRAMMING with BASIC**", SCHAUM'S OUTLINE SERIES IN COMPUTERS, Mc GRAW-HILL BOOK COMPANY, (1982).

7. B. J. Holmes: "**BASIC programming**", Galgotia Book Source, (1983).

8. B. MEEK, P. HEATH and N. RUSHBY (ed.): "**Guide to Good Programming Practice**", Second edition, (1983).

9. C. A. Hofeditz: "**Computer Programming Language Made Simple**", DOUBLEDAY and COMPANY INC., (1984).

10. K. Brenan and S. L. Mandell: "**Introduction to Computer and BASIC programming**", WEST PUBLISHING COMPANY, (1984).

11. D. H. SANDERS: "**COMPUTERS TODAY**", International Students Edition, Second edition, (1985).

12. S. J. Wainwright: "**Programming in MSX BASIC**", Kuma Computers Ltd., (1985).

13. G.G. BITTER: "**COMPUTER LITERACY**", Addison-Wesley Publishing Company, (1986).

14. P. MOOLTON: "**FOUNDATION OF PROGRAMMING THROUGH BASIC**", John Wiley and Sons, (1979).

15. B. ALBRECHT, L. FINKEL and J. BROWN: "**BASIC FOR HOME COMPUTERS**", John Wiley and Sons, (1978).

16. L. Poole and M. Borchers: "**Some Common BASIC programs**", 3rd. Edition, OSBORNE / Mc GRAW-HILL, (1979).

17. H. M. Walker: "**Problems for Computer Solutions Using BASIC**", Winthrop Publishers Inc., (1980).

ملحق جـ :

حلول جميع تمارين الكتاب ما عدا «اجب باختصار»

تمارين (1:

استخدم كلمات قد تعلمتها .

1 ـ (ز)	6 ـ (أ)
2 ـ (ط)	7 ـ (د)
3 ـ (ى)	8 ـ (ح)
4 ـ (ب)	9 ـ (هـ)
5 ـ (و)	10 ـ (جـ)

صح أم خطأ .

1 ـ (صح)	6 ـ (خطأ)
2 ـ (صح)	7 ـ (خطأ)
3 ـ (صح)	8 ـ (خطأ)
4 ـ (صح)	9 ـ (صح)
5 ـ (خطأ)	10 ـ (صح)

اختار الجواب الصحيح .

1 ـ (د)	6 ـ (جـ)

2 - (د)	7 - (ب) و(جـ) و(د)

3 - (د)	8 - (ب)

4 - (ب)	9 - (جـ)

5 - (ب)	10 - (ب)

تمارين (2) :

استخدم كلمات قد تعلمتها .

1 - (ب)	6 - (أ)

2 - (ى)	7 - (ط)

3 - (د)	8 - (جـ)

4 - (ح)	9 - (و)

5 - (هـ)	10 - (ز)

صح أم خطأ

1 - (خطأ)	6 - (خطأ)

2 - (صح)	7 - (خطأ)

3 - (خطأ)	8 - (صح)

4 - (خطأ)	9 - (خطأ)

5 - (صح)	10 - (صح)

اختار الجواب الصحيح .

1 - (جـ)	6 - (جـ)

2 (د)	7 - (ب)

3 - (أ)	8 - (د)

4 ـ (د) 9 ـ (د)

5 ـ (د) 10 ـ (د)

تمارين (3) :

استخدم كلمات قد تعلمتها .

1 ـ (د) 6 ـ (أ)

2 ـ (ز) 7 ـ (هـ)

3 ـ (ب) 8 ـ (ط)

4 ـ (و) 9 ـ (ى)

5 ـ (جـ) 10 ـ (ح)

صح أم خطأ .

1 ـ (خطأ) 6 ـ (خطأ)

2 ـ (خطأ) 7 ـ (صح)

3 ـ (خطأ) 8 ـ (خطأ)

4 ـ (صح) 9 (خطأ)

5 ـ (صح) 10 (خطأ)

اختار الجواب الصحيح .

1 ـ (جـ) 6 ـ (جـ)

2 ـ (أ)، البيانات أكثر من المتغيرات. 7 ـ (جـ)

3 ـ (د) 8 ـ (ب)

4 (ب) 9 ـ (ب)

5 ـ (أ) 10 ـ (أ)

تمارين (4) :

تستخدم كلمات قد تعلمتها

1 - (هـ)	6 - (جـ)
2 - (و)	7 - (ب)
3 - (أ)	8 (ط)
4 - (ى)	9 - (ح)
5 - (د)	10 - (ز)

صح أم خطأ

1 - (خطأ)	6 - (خطأ)
2 - (خطأ)	7 - (خطأ)
3 - (خطأ)	8 - (خطأ)
4 - (خطأ)	9 - (صح)
5 - (خطأ)	10 - (صح)

اختار الجواب الصحيح .

1 - (د)	6 - (د)
2 - (جـ)	7 - (جـ)
3 - (جـ)	8 - (ب)
4 - (ب)	9 - (أ)
5 - (أ)	10 - (د)

تمارين (5) :

استخدم كلمات قد تعلمتها .

1 - (هـ)	6 - (ب)

2 - (د)
3 - (أ)
4 - (ى)
5 (ط)
7 - (ح)
8 - (جـ)
9 - (ز)
10 - (و)

صح أم خطأ

1 - (خطأ)
2 - (خطأ)
3 - (صح)
4 (خطأ)
5 - (خطأ)
6 - (صح)
7 (صح)
8 (صح)
9 - (خطأ)
10 - (خطأ)

اختار الجواب الصحيح .

1 - (د)
2 (جـ)
3 (ب)
4 - (جـ)
5 - (أ)
6 - (جـ)
7 - (د)
8 - (جـ)
9 - (جـ)
10 - (جـ)

الملحق د :

خلاصة عبارات لغة بيسك

```
10 DATA constant, constant, constant,...
20 DEF FN X(variable) = expression
30 DIM dimensioned variable, dimensioned variable,....
999 END
40 FOR variable = expression TO expression STEP expression
50 GOSUB statement number
60 GOTO statement number
70 IF relational expression THEN statement number
80 INPUT variable, variable, variable,...
90 LET variable = expression
100 NEXT variable
110 ON expression GOTO statement number, statement number,...
120 PRINT variable; "STRING"; expression, variable,...
130 READ variable, variable, variable,...
140 REM any explination or comment or remark
150 RESTORE
160 RETURN
170 STOP
```